Commodore
COMPUTER CLUB

65   L. 5.000

La rivista degli utenti di sistemi Commodore

Anno VIII - N. 64 - 25 giugno 1989 - Sped. Abb. Post. Gr. III/70 -CR -Distr.: MePe

Foto di gruppo con zoom programmabile

7 ft.

6 ft.

Tre listati per ingrandire caratteri e immagini col C64

Metti un turbo nell'Amiga

5

Porta utente 64, aprila verso un nuovo mondo

Primo gioco Amiga Basic, labirinti e trabocchetti

65

Commodore COMPUTER CLUB
La rivista degli utenti di sistemi Commodore

Sommario

RUBRICHE

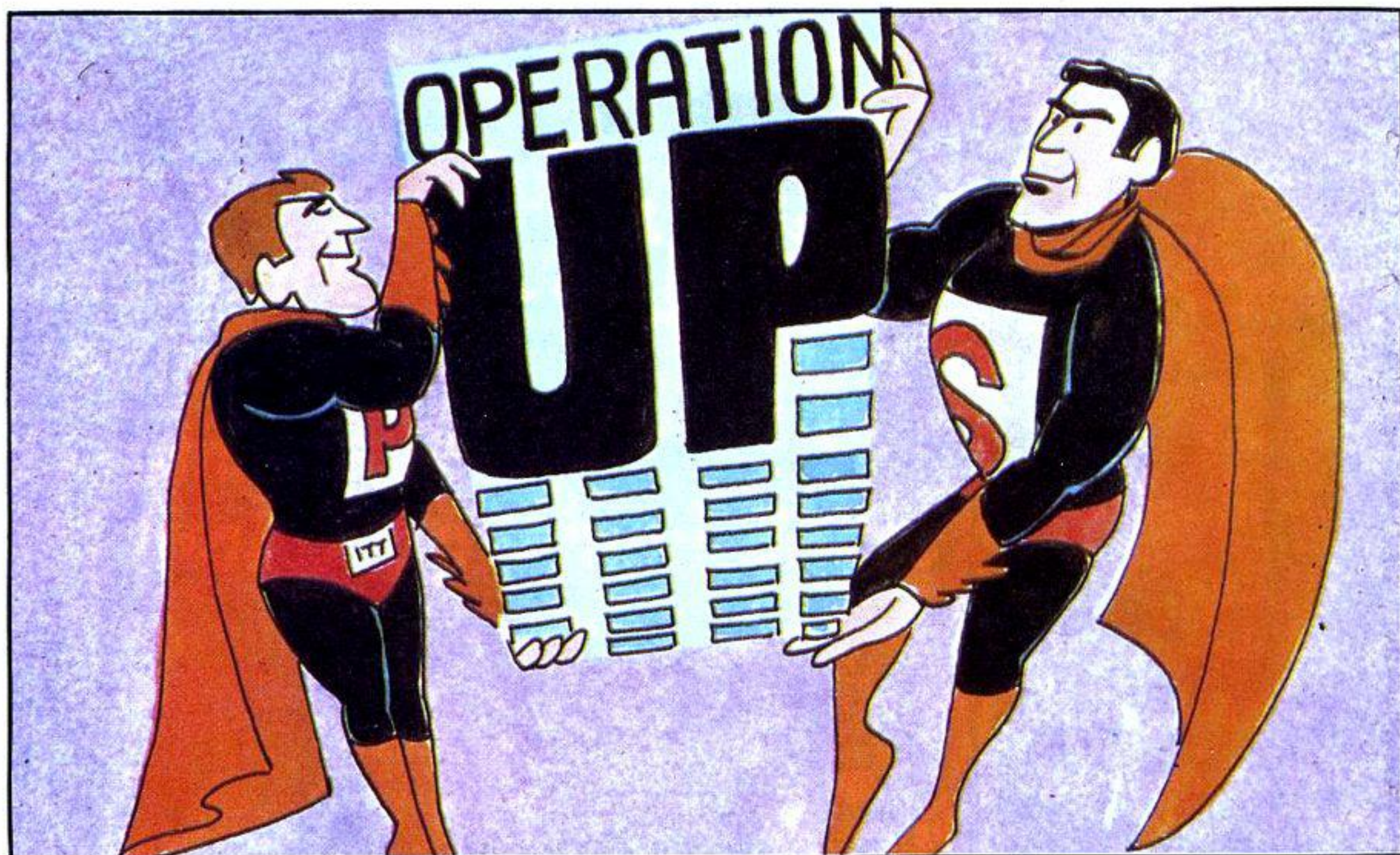
Foto di gruppo con zoom programmabile

Tre listati per ingrandire caratteri e immagini col C64

Metti un turbo nell'Amiga

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Lauria & Miria Colombo, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Domenico Pavone, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma, Franco Rodella, Stefano Simonelli
Grafica: Arturo Ciaglia
Direzione, pubblicità: via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310 - **Redazione:** Tel. 02/5249211
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Angelo Ricupero - Via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Arretrati e software: Lucia Dominioni (Tel. 02/8467348/9 - viale Famagosta 75 - 20142 Milano)
Tariffe: prezzo per copia L. 5.000. Abbonamento annuo (11 fascicoli) L. 50.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 90.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - Jonathan - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - TuttoGatto Videoteca - VR Videoregistrare

L'ANNO CHE VERRA'

E' tempo di bilanci; e di pensare alle prossime, straordinarie(?) vacanze

Nel campo dell'informatica, ed in quello delle pubblicazioni hobbystiche in particolare, il "ciclo" di attività non segue quello solare.

Si avverte, infatti, il risveglio dell'interesse per il settore nei mesi di settembre e ottobre; a novembre, e soprattutto a dicembre, si decide quali periferiche acquistare; a gennaio e a febbraio ci si impegna nell'imparare ad usare un software professionale di rilievo. A marzo e ad aprile vi sono due tendenze opposte tra loro; alcuni tendono al disimpegno, altri si danno da fare per migliorare la loro preparazione o per mettere a punto qualche tecnica appresa in precedenza.

Maggio e giugno, tradizionalmente, costringono gli studenti allo sprint finale per evitare spiacevoli sorprese agli scrutini di fine anno; non c'è dunque molto tempo per il computer (tranne che per gli studenti di informatica, ovviamente!).

Chi dedica tempo al proprio elaboratore nei mesi estivi (e, magari, a partire da maggio) non può che essere l'appassionato evoluto, colui che, insomma, è ormai stato "infettato" dal benefico virus dei chip, del Basic, del L.M. o del Pascal e non intende rinunciare alle piacevolezza che solo un computer è in grado di donare a chi se ne intende.

Parallelamente la nostra pubblicazione, che agli hobbisti (evoluti e non) si rivolge, dedica un minor numero di pagine ai superprincipianti, ma non può far finta di niente di fronte al tepore primaverile, anzi estivo, che inizia a pervadere l'aere.

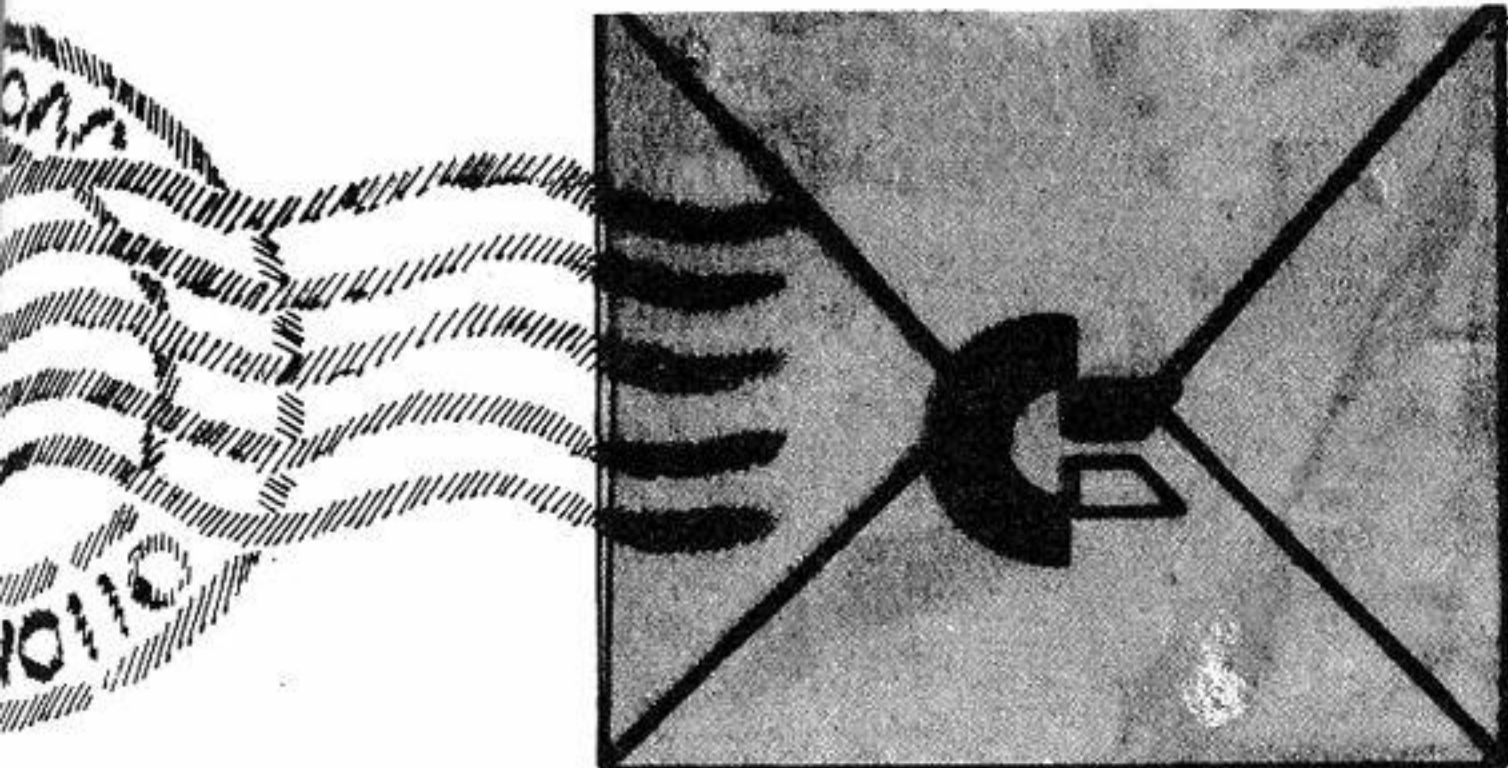
A parte lo squallido tentativo poetico, abbiamo pensato anche a chi, tra una pagina e l'altra, inizia a pregustare il periodo di vacanze che si approssima al galoppo. Tra un articolo sull'assembly ed uno sul basic, tra una dissertazione su Amiga ed una sull'eterno C/64, può esser piacevole affrontare un quiz, pensando che, tra breve, godremo un più che meritato (ma sempre troppo breve) riposo.

Per ciò che riguarda la nostra attività appena svolta, non possiamo che esser contenti, soprattutto se teniamo conto del tono delle lettere pervenute in Redazione.

Sono molti coloro che, grazie a noi, sono passati dal primitivo Print "Pippo" alla gestione di sofisticate tecniche di programmazione. Altri hanno scelto la periferica, o addirittura il nuovo computer.

Spronati dalle sollecitazioni dei lettori affronteremo, tra breve, argomenti più impegnativi, sia su Amiga, sia sul C/128 (a proposito, avete notato che, REALMENTE, teniamo in considerazione le vostre richieste?) sia sul C/64, che non può assolutamente esser considerato "esaurito".

Nel frattempo, chi possiede la raccolta completa degli ultimi fascicoli potrà sfogliarli nuovamente alla ricerca di quel programma da sviluppare, di quella routine da inserire in quell'altro programma che, opportunamente modificato, consente di...

Ci risiamo: eccoci ancora alle prese con listati, dischetti, prompt e pressioni convulse di Run/Stop e Restore; mi viene un dubbio: vuoi vedere che, noi hobbysti, abbiamo il privilegio di essere in continua, eterna vacanza?...

la vostra posta

UN GURU DI TROPPO

□ Vi scrivo per render noto un piccolo(?) bug dell'AmigaDOS...
(Flavio Stanchina - Trento)

• "Provate ad eseguire le seguenti operazioni -continua il nostro lettore- ...

MAKEDIR 1
MAKEDIR 1/2

..e così via, fino a realizzare dieci subdirectory nidificate l'una all'interno dell'altra. Quindi eseguite...

DIR OPT A

Bella la Guru Meditation, vero? L'ho scoperta quasi per caso, esaminando con il suddetto comando la directory di *SideWinder*.

Il bug non dovrebbe creare problemi dal momento che, di solito, si utilizzano poche directory nidificate; ma che cosa può accadere procurandosi un hard disk?

Inoltre volevo dire qualche parola sul comando ASK. Non sono riuscito a trovare le istruzioni su tre manuali dell'Amiga-DOS e su C.C.C. quindi ho dovuto arrangiarmi, e ho scoperto che restituisce errore 5 se si risponde sì e non fa nulla se si risponde no."

Ringraziamo Flavio Stanchina per la segnalazione che giriamo, pari pari, al sig. Commodore (o chi per lui, se esiste). Chissà, forse con la versione 1.5 del Work bench il bug verrà eliminato...

UN BUG PER PAVAROTTI

□ Il programma "Come Pavarotti con il C/64", da voi pubblicato sul N. 63 di Commodore Computer Club, contiene un piccolo errore.
(Anonimo telefonista)

• La riga 860 deve esser modificata. Invece di...
860 For A=1 to WS....
...bisogna digitare:
860 For A=SA to EA
In caso contrario, infatti, verrà sempre stampata l'intera canzone, come si può notare, del resto, studiando le righe 820 e 830.

SALVARE CON AMIGA

□ Con AmigaBasic dovrebbe esser possibile registrare un programma in formato Ascii, ma non riesco assolutamente ad imporre il suffisso ",a" come indicato nel libretto di istruzioni.
(Da alcune telefonate)

• Per registrare un programma Basic è sufficiente la sintassi...
Save "Nome"
In questo modo, però, viene registrato un file che potrà essere decodificato solo usando il linguaggio interprete AmigaBasic. Per sincerarvene, attivate il CLI (da Dos) e tentate di visualizzarne il contenuto, per esempio, con il comando Type:
Type Nome
Verranno visualizzati caratteri privi di senso proprio perchè AmigaBasic ha registrato il programma ricorrendo a codici particolari.

E' però possibile registrare un programma in formato Ascii in modo tale che, appunto, il file possa essere correttamente visualizzato con qualsiasi programma in grado di trattare file Ascii. Tra questi (oltre ai comandi del Dos come Type, Printfiles e così via) ricordiamo i numerosi Word processor e Desk Top Publishing.

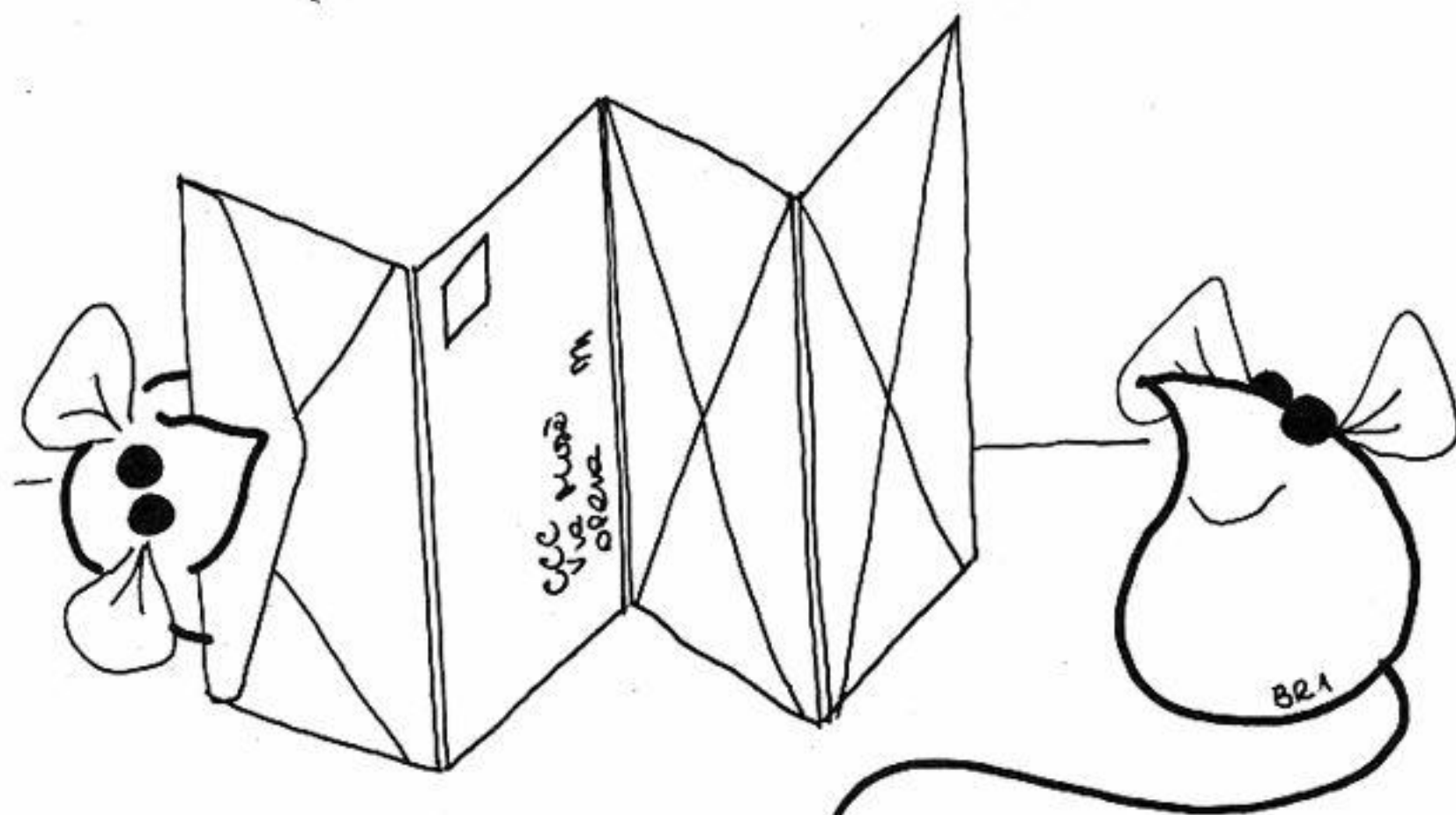
Naturalmente anche un file registrato in formato Ascii potrà essere caricato da AmigaBasic con un normalissimo *Load*. La differenza consiste nel fatto che un file registrato con il suffisso ",a" occupa una maggior quantità di memoria e richiede, di conseguenza, un tempo di caricamento maggiore.

Rimane ora da spiegare come mai i nostri lettori non riescono a registrare un programma dotandolo del suffisso citato.

Ciò è dovuto al fatto che, per effettuare il salvataggio, viene attivato il menu *Project* (ed in particolare, l'opzione *Save* oppure *Save As*) che, per dimenticanza dei progettisti, non consente di apporre suffissi: ciò che viene digitato viene considerato come nome del file.

Per utilizzare le altre forme sintattiche possibili (tra cui l'indicazione di nidificazioni di subdirectory) è necessario cliccare la finestra di output di AmigaBasic e digitare, carattere dopo carattere, l'intero comando, completo di eventuali suffissi:
Save "nome", a
A volte, lo ricordiamo, possono verificarsi inconvenienti non prevedibili utilizzando il menu *Project* per effettuare varie operazioni.

STAMPANTI

□ Vorrei sapere se la stampante che vi indico (*segue indicazione del model-*

***lo*) è realmente compatibile con tutti i computer, che carta usa e che qualità di stampa posso ottenere utilizzandola con un C/64 e con un Amiga.**
(Mario Ciciotti - Avezzano)

• Abbiamo sempre detto che non possiamo dare consigli sugli acquisti da effettuare e non sfuggiamo alla regola nemmeno stavolta.

Per ciò che riguarda la compatibilità ti posso però assicurare (pur senza conoscere il modello in questione) che se la stampante è dotata di interfaccia *Centronics* potrai utilizzarla non solo con Amiga, ma con *qualsiasi* computer Ms-Dos compatibile. La compatibilità con lo standard C/64 (se realmente vantato dall'apparecchio) è garantita solo se sulla confezione è presente la scritta "MPS - 803 compatibile". Altrimenti no.

PROGRESSI

☐ Mi conviene iniziare a studiare il linguaggio macchina del mio C/128 se, quasi certamente, entro un anno acquisterò un Amiga o un computer Ms-Dos compatibile?
(Alessandro P. - Biancavilla)

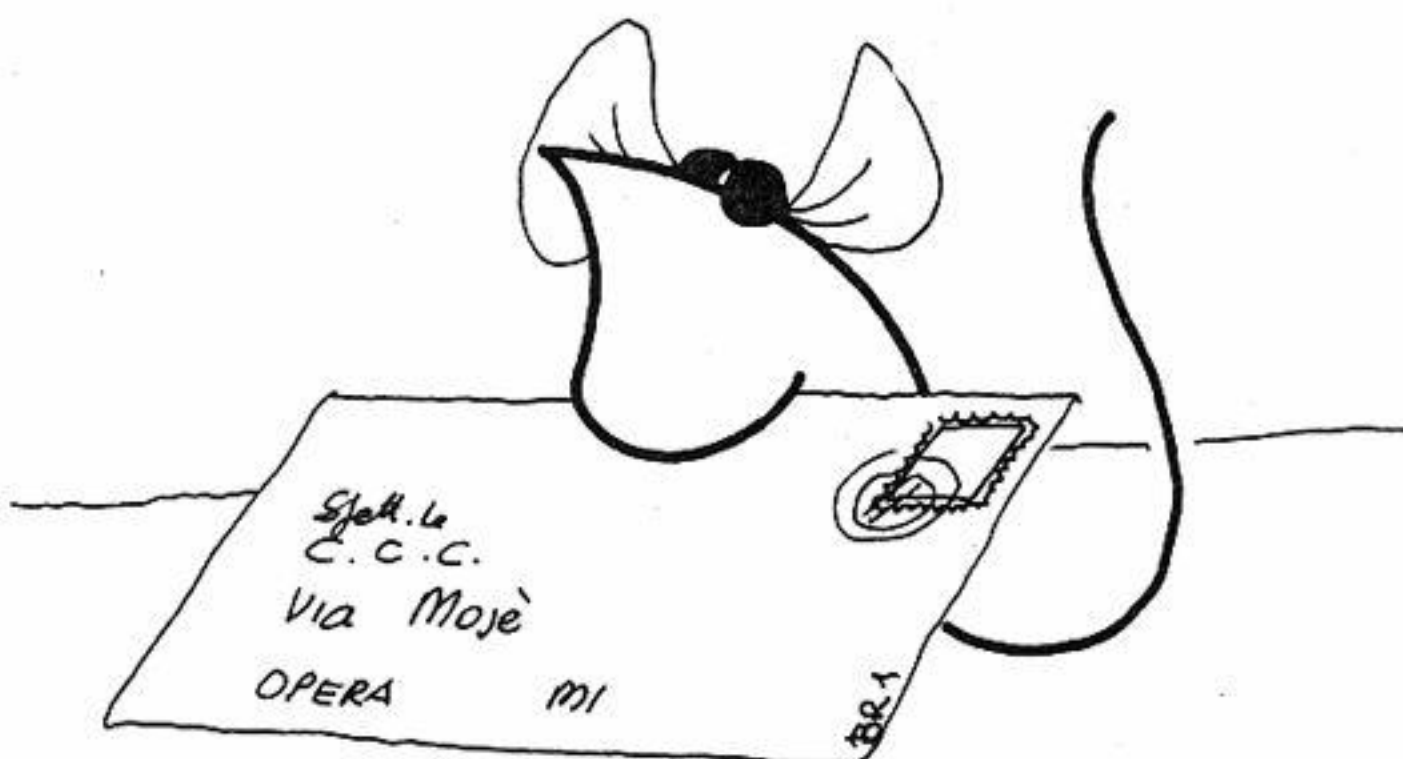
• Lo studio del linguaggio macchina "apre" la mente verso procedure ed algoritmi che nessun linguaggio evoluto riesce a suggerire.

Il modo di operare di un qualsiasi processore costringe ad aver confidenza con puntatori, vettori, istruzioni ad uno o più byte, mappe di memoria, trasferimenti di blocchi di byte, registri interni, utilizzo di routine del sistema e tante altre cose che sono comuni a tutti i microprocessori passati, presenti e futuri.

Semmai, dopo aver cambiato computer, ti accorgerai che lo studio del micro del C/128 si è rivelato utilissimo per comprendere meglio (e più facilmente) le nuove istruzioni del più potente microprocessore con cui avrai a che fare.

AMIGA IN VIDEOGRAFICA

☐ E' possibile digitalizzare immagini provenineti da un VCR privo dell'opzione di fermo-immagine?
(Marco Tofanelli - Orbetello)

• Un'immagine grafica è formata da migliaia di informazioni analogiche che devono esser trasformate in segnali digitali affinchè siano correttamente interpretate, e visualizzate, dal computer. Per svolgere un lavoro del genere è necessario parecchio

IN ATTESA DI GIUDIZIO

RASTER POCO CHIARO
(Fabio Moretti - Torino)

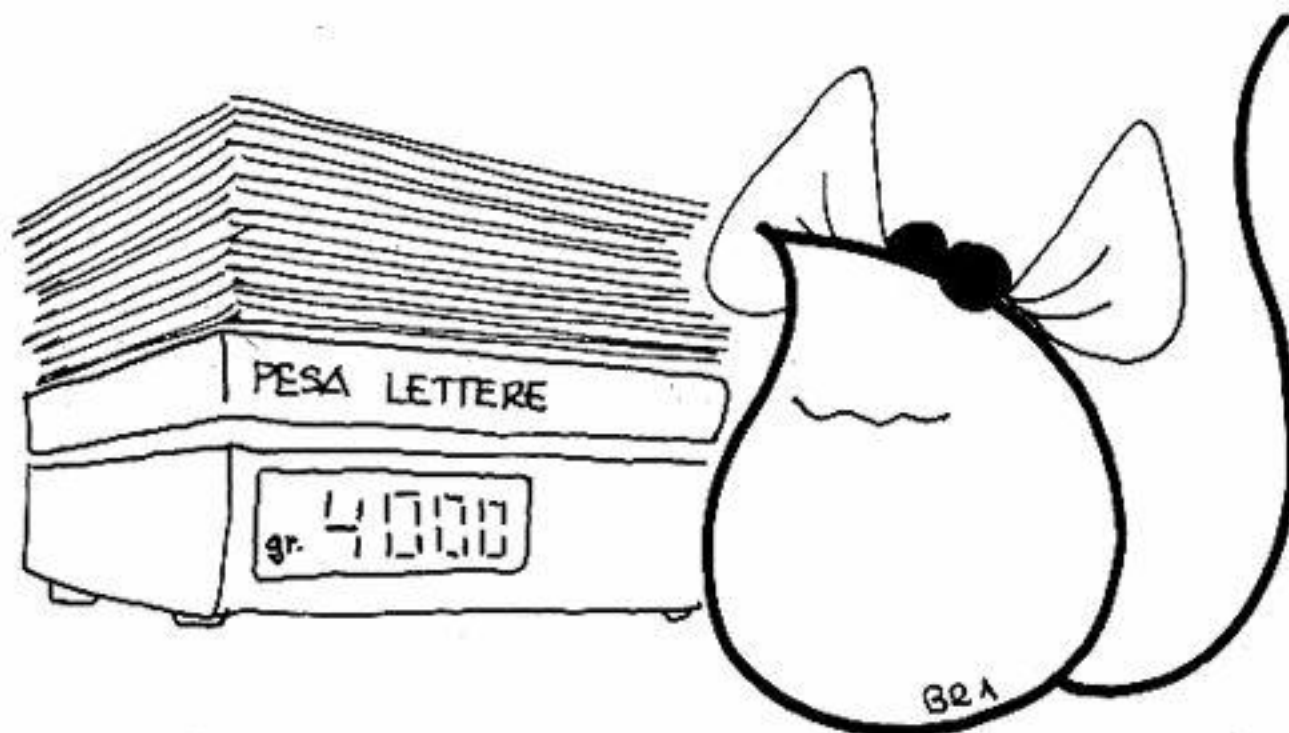
I sei brevi programmi che hai inviato, e che rappresentano una raccolta di simpatici effetti ottenibili manipolando il Raster Register del C/64, sono molto interessanti (e più che degni di esser pubblicati) ma, purtroppo, sono privi del disassemblato *commentato* che si rivela indispensabile ai lettori che vogliono saperne di più sul l.m. Hai invece inviato il solo disassemblato, privo di qualsiasi commento!
Inoltre hai dimenticato di includere il listato caricatore, in Basic, che si occupa di allocare in memoria la parte "comune" alle sei routine. Per finire, usi *Easy Script* in modo improprio. Perchè non impari ad usarlo al massimo delle sue possibilità? Ti faccio i complimenti, comunque, per l'impegno che hai dimostrato nell'affrontare lo studio del raster register.

UN CONTO CORRENTE PER AMIGA
(Emiliano Sciarra - Civitavecchia)

Il programma, pur se curato nell'aspetto estetico, è privo di commento "incorporato" e non mi sembra tanto intuitivo da giustificare l'assenza di un *help* in linea. Inoltre il cursore non lampeggia e non si riesce ad individuare il punto esatto in cui verrà visualizzato ciò che stiamo per digitare. Manca, poi, l'opzione di correzione immediata: si è costretti a digitare per intero il rigo di immissione movimenti, prima di correggere il dato errato; ma anche in questo caso è necessario digitare nuovamente anche i dati esatti.

La gestione delle varie cifre in gioco risulta macchinosa, non è possibile correggere errori di precedenti annotazioni (tranne l'ultima) nè limitarsi a visualizzare l'estratto conto sul video, ma solo su stampante. Il programma, comunque, può essere considerato sufficientemente valido, almeno se viene considerato come prima versione di un più impegnativo lavoro.

PS DOS PER C/64
(Stefano Petrullo - S. G. Milanese)

La raccolta di programmi di utilità che hai inviato (da attivare grazie ad un menu

di tua realizzazione) è piuttosto ampia ma, come tu stesso ammetti, non è possibile "rientrare" nel menu principale al termine delle operazioni. Olcune routine, in Basic, meriterebbero di esser migliorate inserendo almeno un controllo sulla presenza di un file su disco o sulla effettiva presenza della periferica su cui agire. In caso contrario, infatti, il programma si interrompe con segnalazione di errore.

CACCIA AL BIS

(Daniele Rondina)

Purtroppo il dischetto risulta danneggiato e non ci è possibile leggere il contenuto.

VOCABOLARIO

(Donato Donati - Ancona)

La tecnica che hai usato per archiviare i vocaboli è nota da tempo, ma risulta tuttavia molto valida e consente una ricerca agevole e veloce. Il programma, quindi, è piuttosto interessante anche se, con i suoi 66 blocchi (corrispondenti a circa 30 schermate di listato!) è troppo lungo per esser pubblicato. Ricordo, comunque, che un *vero* vocabolario non serve solo per ricercare la traduzione di un termine, oltre ad eventuali sinonimi, ma viene consultato per individuare frasi idiomatiche, pronuncia ed altre "notizie" più o meno interessanti.

Torno a ripetere che un C/64 (pur se dotato di drive) non può risolvere il problema. Occorre, come minimo, un computer professionale dotato di lettore laser CD-Rom!

WINDOWS DREAM

(Nicola Ghelli - Villañova)

Il Sistema Operativo per C/64 che hai inviato è veramente interessante, pur se di applicazioni limitate. Si vede che conosci il linguaggio macchina molto bene e che ti piace curare anche i particolari più reconditi. Come tu stesso hai ammesso, tuttavia, il lavoro non può considerarsi completo e, anche se lo fosse, probabilmente non tutti sarebbero in grado di apprezzarlo per quello che vale. Una proposta: dal momento che sei così bravo, prova a scrivere un videogame, interamente in l.m. che sia davvero interessante per l'altra nostra pubblicazione su disco ("Commodore Club"): poi fatti vivo nuovamente.

tempo che il computer (o la scheda specifica) impiega per decodificare l'immagine ed i colori in cui è composta. Durante questo periodo di tempo l'immagine, ovviamente, deve essere immobile, pena la distorsione cromatica o, addirittura, la deformazione dell'immagine stessa.

Naturalmente sono in commercio schede di basso costo (che richiedono un lungo periodo di elaborazione e, quindi, il fermo immagine di un VCR o una telecamera fissa su un soggetto immobile) e schede di elevato livello tecnologico (e prezzo!) che consentono, tuttavia, di operare ad alta velocità e di digitalizzare, di conseguenza, anche fotogrammi appartenenti a sequenze in rapido movimento.

Bisogna quindi fare i propri conti: è meglio cambiare VCR (con uno dotato di fermo immagine) oppure cambiare digitalizzatore (ed utilizzare il vecchio VCR)?.

C'è un'ultima possibilità, a patto di disporre di una telecamera: porre davanti all'obiettivo una fotografia (magari scattata con 1/1000 di secondo!) e digitalizzarla...

C'E', MA NON SI VEDE

☐ Digitando *List*, dopo aver caricato alcuni programmi, sembra che questi non siano presenti in memoria. Eppure, durante il caricamento, non compare il messaggio *Loading error* che, di solito, viene emesso dal C/64 in casi del genere.

(Giandavide Alfano - Palermo)

• Il comando *List* è un comando del Basic ed agisce, appunto, solo se in memoria è presente un programma Basic a partire, per di più, da una ben precisa locazione di memoria. In tutti gli altri casi il computer ritiene che non esistano listati da... listare.

Alcuni programmi commercializzati sono allocati (: memorizzati) a partire da locazioni di memoria diverse da quelle standard e, di conseguenza, *invisibili* al comando List; tali files sono, di solito, programmi in linguaggio macchina, schermate grafiche o musiche digitalizzate che verranno in seguito attivate secondo procedure che variano caso per caso.

Pertanto il comando *Load*, impartito per caricare tali files, funziona correttamente, e in modo totalmente automatico, utilizzando il registratore a cassette. Con il drive, invece, gli stessi files devono essere caricati con il suffisso ", 8, 1"; in caso contrario il comando List può visualizzare caratteri incomprensibili.

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

Per accelerare il servizio, ricordate di indicare sempre la data di spedizione dal momento che questa costituisce diritto di precedenza.

A tutt'oggi (con la pubblicazione del presente fascicolo) risulta evasa l'intera corrispondenza pervenuta in Redazione entro la fine del mese appena trascorso.

Non ci è possibile dare consigli su acquisti da effettuare nè esprimere pareri estranei alla logica della correttezza professionale e commerciale.

AMIGA, 64, DATASSETTE E 1581

□ Sul N. 60 di C.C.C. avete recensito il programma "C/64 Emulator" che consente di collegare l'Amiga al drive 1541 in modo da utilizzare il software del C/64. Perchè non realizzano qualcosa del genere per connettere, invece del drive, il registratore a cassette?
(Alessandro Galluzzo - Messina)

□ Perchè non pubblicate un fascicolo "Speciale drive" specifico per il 1581?
(Sebastiano Tringali - Augusta)

• Dobbiamo metterci in testa, una buona volta, che la realtà *hobbistica* è totalmente diversa dalla realtà *commerciale*.

Se una software house decide di impegnarsi nella stesura di un programma, non si chiede se questo sarà bello, ma solo se riuscirà a vendere un adeguato numero di copie in modo da ottenere vantaggi economici. Tale ragionamento, che può apparire brutale ma che è alla base di una sana economia, è applicato a qualsiasi campo dell'attività umana: la Fiat, ad esempio, non avrebbe speso *miliardi* per progettare, sperimentare e pubblicizzare la Tipo se non fosse stata sicura di vendere macchine.

Tornando a noi, le domande che si sono posti i produttori di *C/64 Emulator* sono state le seguenti:

• quante confezioni saranno acquistate da possessori di drive e quante da possessori di datassette?

• che impegno è necessario per realizzare un emulatore per registratore o per drive?

• quanto costa produrre, nei due casi, l'interfaccia da allegare alla confezione di *C/64 Emulator*?

• quale è l'affidabilità dei due sistemi?

Le risposte ottenute (completate, magari, da una ricerca di mercato) ha portato alla conclusione che l'unica cosa saggia da fare (in termini commerciali) era di privilegiare il drive e non il registratore.

Nulla vieta, però, che il fenomeno sia ridimensionato e che, magari dopo il successo di vendita, decidano di andare incontro anche ai possessori del solo datassette, proponendo un prodotto specifico.

Allo stesso modo, la pubblicazione di un fascicolo speciale dedicato esclusivamente al drive 1581 risulterebbe un vero disastro economico: il numero di drive di questo tipo attualmente in circolazione non giustificherebbe sforzi editoriali per finanziare iniziative del tipo suggerito. Nemmeno per convertire programmi, già pubblicati, dallo standard 1541 a quello 1581.

PERICOLO HARDWARE

□ Vorrei utilizzare la porta utente (o quella del registratore a cassette) per comandare apparecchi che funzionano, però, a 220 volt. Usando relè, il pericolo di corti circuiti è scongiurato?
(M. Cibelli - Salerno)

• La porta utente non fa altro che "presentare" 5 volt (oppure zero volt) su uno degli otto connettori specifici (vedi numeri precedenti di C.C.C.). Se non bastasse, la corrente disponibile è di pochi milliampere, appena sufficienti ad accendere un LED, a comandare un circuito integrato TTL (del tipo, per intenderci, SN XXXX), oppure micro relè a basso assorbimento. Per evitare pericoli di vario genere è indispensabile essere piuttosto esperti in montaggi elettronici ed utilizzare, per la massima sicurezza, particolari dispositivi opto elettronici.

Questi, in pratica, funzionano su un principio semplicissimo: da una parte è presente un LED, da accendere o spegnere con alimentazioni relativamente basse (leggi: porta utente). L'altra estremità del componente è costituita da un sensore luminoso, posto in prossimità del LED ed isolato con l'esterno, che, a sua volta, viene collegato al circuito da comandare. Dal momento che il LED ed il sensore, benchè vicini fisicamente, sono *totalmente* isolati tra loro, in caso di corto circuito si brucia il sensore e non il LED, o, peggio, la porta utente con esso "solidale".

Naturalmente il circuito stampato destinato a contenere il progetto deve essere realizzato con cura particolare, in modo da scongiurare qualsiasi altro pericolo.

P.S. Non mi sembra che il N. 63 sia poi così brutto come dici nella lettera. La prossima volta, comunque, specifica gli articoli che non ti sono piaciuti: e soprattutto il perchè.

I DRIVE DI AMIGA

□ Posseggo un Amiga 500 dotato del solo drive interno. Per esaminare la directory di un disco dovrei (da CLI) digitare *Dir*, ma, se lo faccio, compare la richiesta di inserimento del disco di Workbench. Se, però, inserisco quest'ultimo disco, viene visualizzata la *sua* directory e non quella del disco

OPINIONI

La rubrica "La voce dei lettori" è nata per dare spazio (e sfogo...) ai lettori che, pur non avendo nulla di particolare da chiedere, intendono far sentire la propria voce, chiedere solidarietà ad altri utenti o, semplicemente, riportare osservazioni ritenute utili o importanti. Per questioni di spazio le lettere sono spesso condensate; vengono poi "censurate" le osservazioni ed i termini troppo "forti" che possano arrecare offesa ai lettori, agli inserzionisti e... a noi stessi. In calce viene riportata la sola iniziale del cognome del lettore (salvo casi particolari), per fare in modo che questi possa sentirsi ancor più libero di esprimere la propria opinione. Coloro che desiderano il minor numero di "tagli" possibile alle loro missive sono pregati di inviarle su DISCO, usando il word processor Easy Script.

che vorrei esaminare. Come posso fare?
(Michele - Firenze)

• Il sistema operativo di Amiga consente di individuare un disco con due metodi: mediante indicazione del *drive* (che si suppone contenga un disco) e mediante indicazione del *nome* del disco stesso. Purtroppo, però, nessun comando del DOS di Amiga è residente e ne consegue che, ogni volta che si impartisce un comando, Amiga è costretta a caricarlo da Workbench per eseguirlo.

Nel caso segnalato dal nostro lettore, pertanto, il comando *Dir* (oppure *Dir Df0:*) viene "applicato" al drive interno; purtroppo, al momento dell'esecuzione di Dir, nel drive interno è presente il dischetto di Workbench e viene quindi visualizzata la sua directory!

Per ovviare all'inconveniente i metodi possibili sono due. Il primo consiste nel rendere residenti alcuni comandi (tra cui Dir) in modo che venga soppressa la richiesta di inserimento di Workbench.

Il secondo metodo, più semplice, consiste nell'assegnare il nome del dischetto su cui operare. Se, ad esempio, il dischetto ha nome "Prova", si dovrà digitare...

Dir Prova:

...non dimenticando il carattere di doppio punto (:) finale che consente di identificare il termine "Prova" come *periferica* e non come un nome qualsiasi.

In quest'ultimo caso Amiga chiederà, come prima, l'inserimento di Workbench; ma, una volta caricato in memoria il comando, chiederà la sostituzione di Workbench con il disco di nome "Prova".

E se non conosciamo il nome del dischetto? Bè, è sufficiente uscire da CLI (comando: *Endcli*) ed esaminare l'icona del dischetto incriminato; poi, ritornare in ambiente Dos.

Analoghe operazioni si devono compiere per altri comandi del Dos (Copy, List e così via).

La possibilità di utilizzare il nome del disco o quello della periferica rappresenta una novità per ciò che riguarda i sistemi operativi. In ambiente Ms-Dos, ad esempio, il nome del disco non viene utilizzato in alcun modo.

LA VOCE DEI LETTORI

MOUSE IMPRECISO

(Francesco Mascetti - Roma)

Il mouse del mio Amiga presentava notevoli imprecisioni di spostamento ai quali ho ovviato grazie ad un lavoro seplice, ma che richiede pazienza.

Se dovesse capitare anche a voi, munitevi di un po' di filo di cotone sottile; il problema può esser causato da una deformazione dei fermi che mantengono stabili gli assi dei cilindretti del mouse (vedi, in C.C.C. N. 63, la risposta ad una lettera). Tale deformazione causa movimenti inopportuni ai cilindretti stessi che, spostandosi di poco avanti e indietro, diventano imprecisi. Basta quindi realizzare un cappio con il filo intorno al fermo e fissare le estremità del filo ad una vite, tirando i capi tanto da far accostare saldamente il fermo all'asse del cilindretto per eliminare il gioco indesiderato. Purtroppo il rimedio è un po' spartano, ma evita l'acquisto di un nuovo mouse.

La Commodore, comunque, potrebbe provvedere a costruire accessori più affidabili.

DRIVE E MALOCCHIO

☐ **Aiuto! Il mio C/128, dotato di drive 1541-C, carica qualsiasi programma in modo 128, ma solo il 30% di software in modo 64. Ho già provato allineatori e programmi di ogni tipo: il drive risulta perfettamente a posto. Incomincio a pensare al malocchio...**
(Adriano Locci)

• Se i dischetti che non riesci a caricare sono privi di difetti (cioè funzionano perfettamente sui computer di tuoi amici) non c'è altro motivo se non quello che sospetti.

In questo caso, il venerdì dopo il tramonto, indossa una veste rossa o, in alternativa, veruna. Quindi appoggia una foto di Amiga sul dischetto indemoniato e, aiutandoti con la mano sinistra, compi l'atavico gesto dell'ombrello toccando con violenza il braccio destro e pronunciando la magica parola *Tiè*.

Se non dovesse funzionare, prova con il registratore...

MINI BUG MUSICALE

☐ **Mi sono accorto che il mio programma da voi pubblicato ("Oltre la musica del SID", C.C.C. N. 61) contiene un piccolo errore.**
(Moreno Paisi - Pioltello)

• "Nel listato che si occupa delle modifiche per il C/64 (pagina X dell'inserto) la riga 330 deve essere sostituita con la seguente:

```
330 CO=32: RG=Z: Gosub 440: Print "
": RG=Y: Gosub 440: Print Chr$(95):
Z=Y
```

Mi dispiace dell'imprecisione e chiedo scusa ai lettori per il mancato funzionamento del listato".

Un piccolo errore può capitare a chiunque.

Del resto i lettori più esperti, ai quali era destinato l'articolo in questione, si erano accorti da soli del mini-bug.

STAMPASCHEDINE

☐ **Ho molto apprezzato le vostre pubblicazioni relative al software per il Totocalcio. Perchè non pubblicate una routine che, dopo aver elaborato i pronostici, stampi anche le corrispondenti schedine?**
(Rosario Casarella - Liscate)

• Il suggerimento del nostro lettore è sicuramente valido ma, purtroppo, si scontra contro due insormontabili ostacoli.

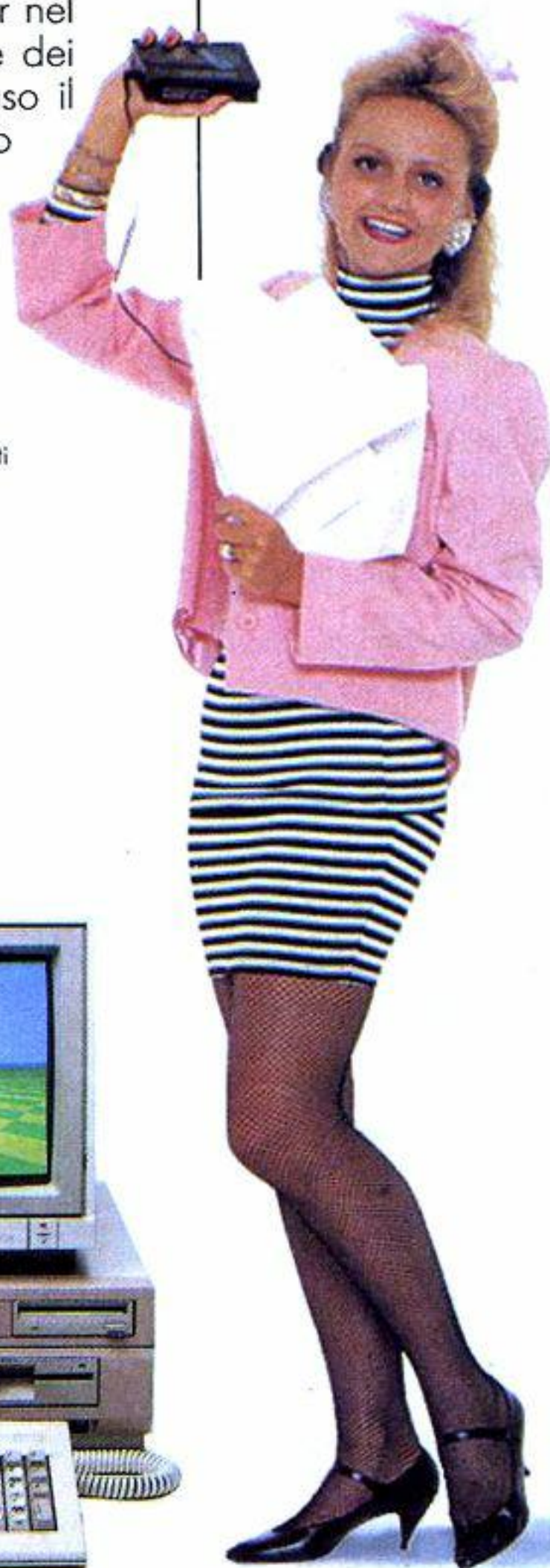
Il primo riguarda proprio le schedine. A differenza del *Totip* che fornisce i moduli

Amiga 2000

Amiga 2000 è il collaboratore ideale per preparare facilmente ed in pochissimo tempo al video testi e grafici e riportarli poi su carta, su lucidi o diapositive. È uno strumento creativo semplice e immediato per realizzare soggetti fantastici in due o tre dimensioni con 4.096 colori. Qualunque sia la tua professione, con Amiga 2000 hai a disposizione le meraviglie dell'ambiente Amiga DOS e del sistema MS-DOS, con i quali potrai elaborare i tuoi progetti con una grafica ineguagliabile. In più, Amiga 2000 è Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto da Commodore, che può così vantare la più lunga esperienza nel settore.

- MS-DOS e XENIX sono marchi registrati della Microsoft Inc.
- UNIX è un marchio registrato della AT&T.
- Commodore e Amiga sono marchi registrati della Commodore Inc.

Amiga 2000.

Per te che vuoi un amico sulla scrivania.

Amiga 2000.

Per te che vuoi lavorare con un creativo alla tua altezza.

Amiga 2000

Per te e per i tuoi progetti che non hanno bisogno solo del sistema MS-DOS.

C'è un Commodore p

PC 10.

Per te
che vuoi
un personal
che faccia
scuola.

PC 40.

Per te
che vuoi lavorare
solo con
i numeri uno.

PC 60.

Per te
che vuoi
una gamma
completa
di soluzioni
al servizio
della tua
Azienda.

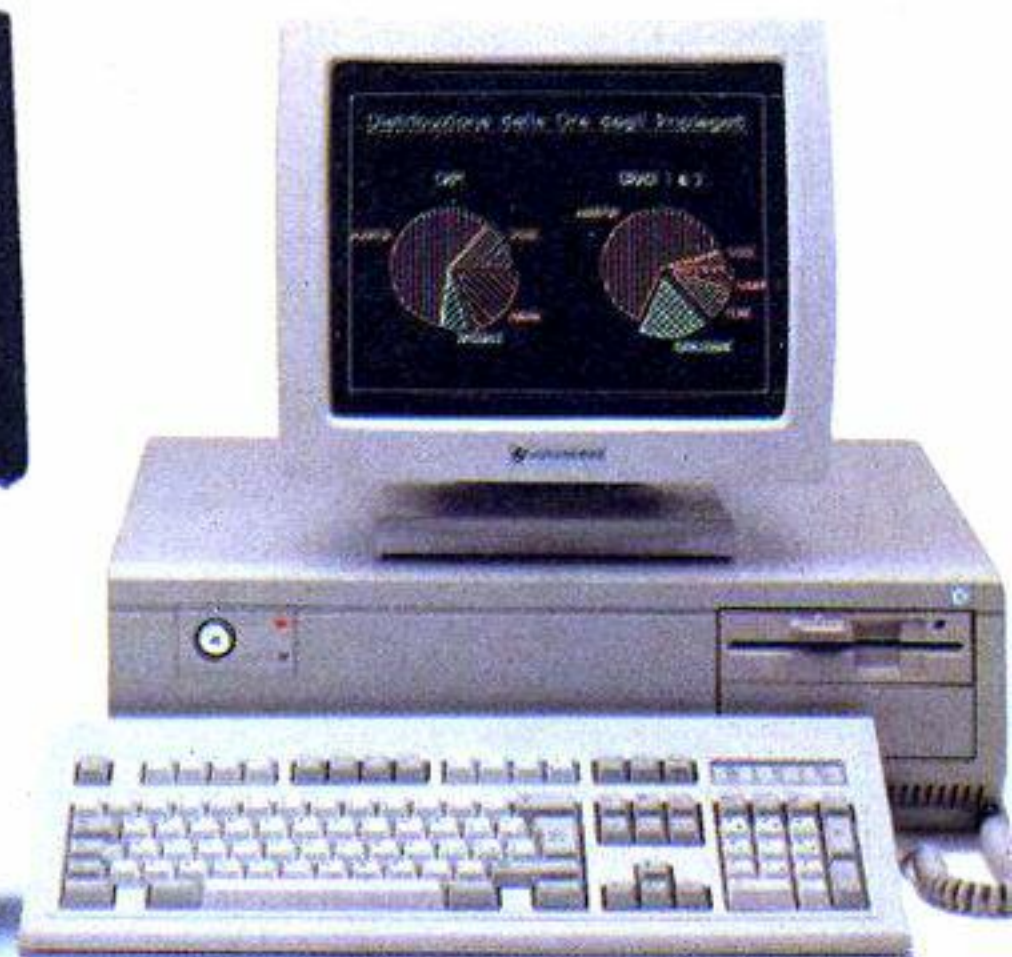
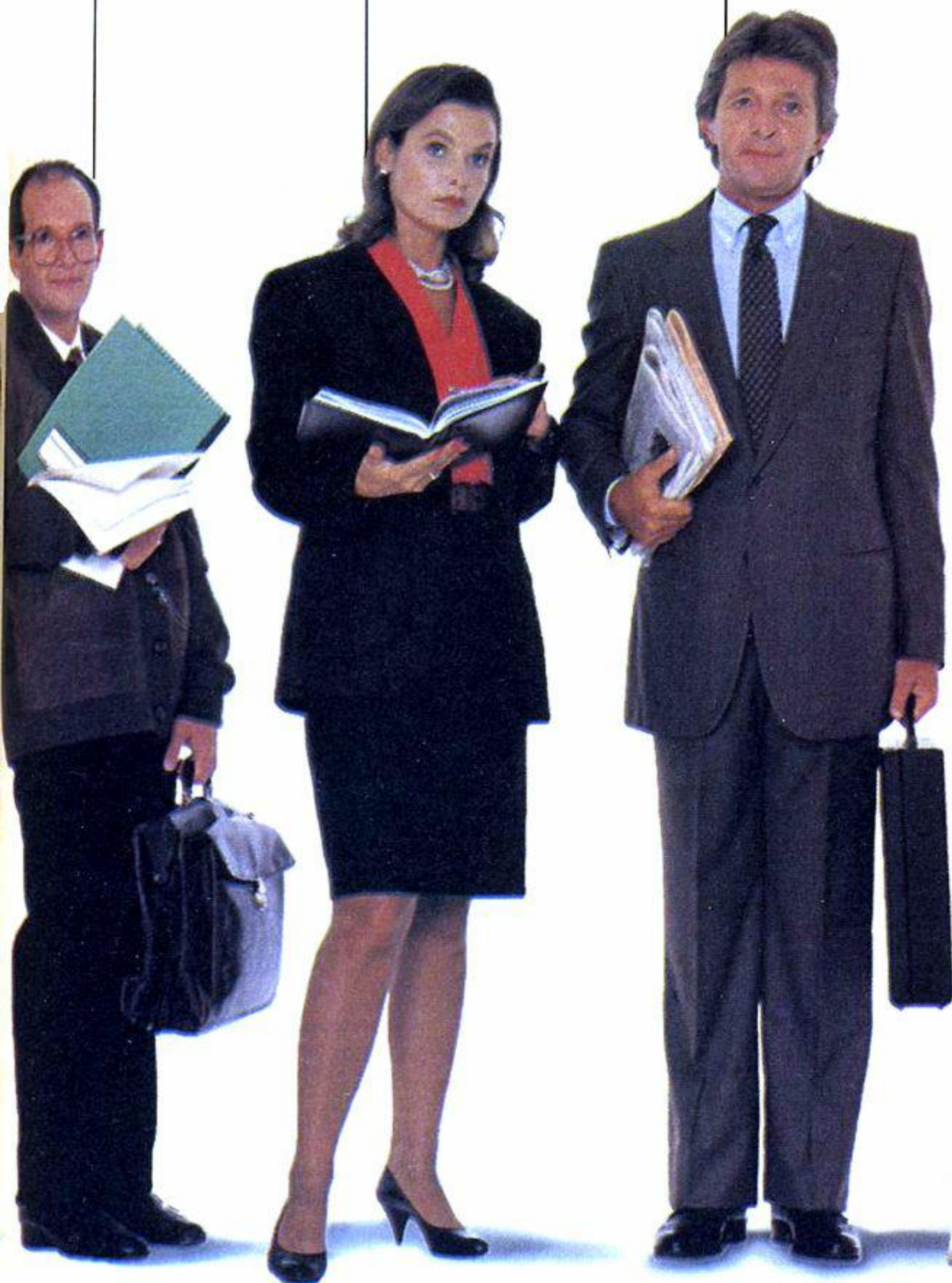
PC

La linea PC Commodore garantisce nel mondo MS-DOS – grazie ai modelli XT, AT e 386 – la soluzione ad ogni problema. Il PC10 (XT) e PC20 (XT) rappresentano la soluzione ideale di partenza. I modelli PC40 (AT) assicurano la potenza e l'espandibilità necessarie a soddisfare ogni esigenza di crescita in qualunque settore. Il PC60 (386) – grazie alla sua potente architettura – è il top della gamma. Sui modelli PC40 e PC60 potrai installare anche il sistema operativo Xenix® o Unix® se la tua azienda lo richiede, e potrai inoltre collegarli in rete. In più PC10, 20, 40, 60 sono Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto dalla Commodore, che può così vantare la più lunga esperienza nel settore.

Per informazioni, dalle 14 alle 17,30
Hot-Line 02-61832245/6

...er te, chiunque tu sia.

idea 2

Concessionario
Commodore
Sistemi Professionali

Lombardia

Milano
- A'ION - Via Bigli, 11
- AL RISPARMIO - V.le Monza, 204
- B.C.S. - Via Montegani, 11
- BRAHA ALBERTO - Via Pier Capponi, 5
- EDI CARED - Via Pietro Calvi, 20
- E.D.S. - C.so Porta Ticinese, 4
- FLOPPERIA - V.le Montenero, 31
- GI-SETTANTA - Via Burlamacchi, 4
- GIGLIONI - V.le Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCUCCI - Via F.lli Bronzetti, 37
- NEWEL - Via Mac Mahon, 75
- SUPERGAMES - Via Vitruvio, 38
- 68000 E DINTORNI - Via Washington, 91

Provincia di Milano
- ALL COMPUTER - Residenza Sassi, 312 - Milano 3 - Basiglio
- PENATI - Via Verdi, 28/30 - Corbetta
- S.A.T. - Via Milano, 24 - Nerviano
- IL CURSORE - Via Campo dei Fiori, 35 - Novate Milanese

Bergamo
- DERCOM - Via Borgo Palazzo, 65/A

Provincia di Bergamo
- COMPUTER SHOP - Via V. Veneto, 9 - Capriate San Gervasio

Brescia
- MASTER INFORMATICA - Via F.lli Ugoni, 10/B

Provincia di Brescia
- CAVALLI PIETRO - Via X Giornate, 14/B - Castrezzato
- DATA SYSTEM NEW - Via Gramsci, 33 - Concesio
- MEGABYTE - P.zza Maluezzi, 14 - Desenzano del Garda

Provincia di Como
- CIMA ELETTRONICA - Via Leonardo da Vinci, 7 - Lecco
- FUMAGALLI - Via Cairoli, 48 - Lecco
- RIGHI ELETTRONICA - Via Leopardi, 26 - Olgiate Comasco

Cremona
- MONDO COMPUTER - Via Giuseppina, 11/B
- PRISMA - Via Buoso da Dovara, 8

Provincia di Cremona
- EUROELETTRONICA - Via XX Settembre, 92/A - Crema

Provincia di Mantova
- CLICK - ON COMPUTER - S.S. Goietese, 168 - Goito

Pavia
- POLIWARE - C.so Carlo Alberto, 76

Provincia di Pavia
- LOGICA MAINT - Via Montegrappa, 32 - Vigevano

Provincia di Sondrio
- FOTONOVA - Via Valeriana, 1 - San Pietro di Berbenno

Varese
- IL CENTRO ELETTRONICO - Via Morazzone, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese
- CURIO TRE - Via Ronchetti, 71 - Cavaria
- J.A.C. NUOVE TECNOLOGIE - C.so Matteotti, 38 - Sesto Calende

Piemonte

Cuneo
- ROSSI COMPUTERS - C.so Nizza, 42

Provincia di Cuneo
- PUNTO BIT - C.so Langhe, 26/C - Alba

Novara
- SOFTEAM - Via Locchi, 6

Provincia di Novara
- L.A.E. SOFTWARE - C.so Cavour, 46/59 - Arona
- ALL COMPUTER - C.so Garibaldi, 106 - Borgomane- ro
- ELLIOTT COMPUTER SHOP - Via Don Minzoni, 32 - Intra

Torino
- ALEX COMPUTERS - C.so Francia, 233/4
- DE BUG COMPUTER - C.so Vittorio Emanuele II, 22
- DESME UNIVERSAL - Via San Secondo, 95
- IL COMPUTER - Via Nicola Fabrizi, 126
- STAF - C.so Regina Margherita, 97

Provincia di Torino
- DIAM INFORMATICA - C.so Francia, 146 BIS - Cascine Vica - Rivoli

Provincia di Vercelli
- C.S.I. TEOREMA - Via Losana, 9 - Biella
- CHIP - Via Martiri della Libertà, 47 - Cossato

Veneto

Belluno
- UP TO DATE - Via Vittorio Veneto, 43

Padova
- SARTO COMPUTER - Via Armistizio, 79

Trentino Alto Adige

Bolzano
- COMPUTER POINT - Via Roma, 82

Provincia di Bolzano
- ELEKTRO TAPPEINER - P.zza Principale 90 - Silandro

Trento
- CRONST - Via G. Galilei, 25

Friuli Venezia Giulia

Pordenone
- SIEL - Via Colonna, 45

Udine
- CO.R.EL. ITALIANA - Via Tavagnacco, 91
- MOFERT 2 - Via Leopardi, 21

Liguria

Genova
- ODEL - Via Orsini, 4 R
- SALS INFORMATICA - Via G. D'Annunzio, 2

Emilia

Bologna
- MINNELLA ALTA FEDELTÀ - Via Mazzini, 146/2
- SINOPIA INFORMATICA - V.le Pietramellara, 11

Provincia di Bologna
- S.C. COMPUTERS - Via Enrico Fermi, 4 - Castel San Pietro
- S.P.E. INFORMATICA - Via di Mezzo Ponente, 385 - Crevalcore

Modena
- VIDEO VAL WILLY COMPUTERS - Via Canaletto, 223

Provincia di Modena
- NEW MEDIA SYSTEM - Via Roma, 281 - Soliera

Parma
- NEW LIST COMPUTER - Via Nazario Sauro, 9

Reggio Emilia
- COMPUTERLINE - Via San Rocco, 10/C
- POOL SHOP - Via Emilia S. Stefano, 9/C

Romagna

Repubblica di San Marino
- A.C.S. - Via Nonagualdaria - Cailungo

Ferrara
- BUSINESS POINT - Via Carlo Mayer, 85

Provincia di Forlì
- TOP BIT - Via Veneto, 12 - Forlimpopoli
- EASY COMPUTER - Via Lagomaggio, 50 - Rimini
- NUMERO - Via Battaglini, 21 - Rimini

Provincia di Ravenna
- E.T.S. - Via Saffi, 1 - Alfonsine
- P.L.Z. INFORMATICA - P.zza Sercognani, 6 - Faenza

Toscana

Arezzo
- DELTA SYSTEM - Via Piave, 13

Firenze
- M.T.S. DISTRIBUZIONE - Via di Novoli, 64 A
- TELEINFORMATICA TOSCANA - Via Bronzino, 36

Grosseto
- COMPUTER SERVICE - Via Dell'Unione, 2

Livorno
- ETA BETA COMPUTER - Via S. Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Lucca
- IL COMPUTER - V.le Colombo, 216 - Lido di Camaiore

Pisa
- ELECTRONIC SERVICE - Via della Vecchia Tranvia, 10

Pistoia
- ELECTRONIC SHOP - Via degli Scalzi, 3

Siena
- RENATO BROGI - P.zza Gramsci, 28

Provincia di Siena
- ELETTRONICA di BIFOLCHI - Via di Gracciano nel Corso, 111 - Montepulciano
- ELETTROMERCATO di BURRINI - Via Toscana, 6 - Monteriggioni

Umbria

Perugia
- MIGLIORATI - Via S. Ercolano, 3
- STUDIO SYSTEM - Via R. D'Andreotto, 49/51

Lazio

Roma
- D.R.R. - Via Giovanni Giorgi, 6

Abruzzo

Teramo
- MAX CENTER - V.le Crispi, 68

Puglia

Bari
- ARTEL - Via Guido d'Orso, 9
- COMPUTER'S ARTS - V.le Meucci, 12/B
- ELETTRONICA SISTEMI - V.le della Repubblica, 67/69
- PAULICELLI SABINO & FIGLI - Via Fanelli, 231/C

Campania

Napoli
- DARVIN - Calata San Marco, 25
- DPC INFORMATICA - Via E. Nicolardi, 129
- GENERAL COMPUTERS - Via Bernini, 101
- SPY - Via Domenico Fontana, 135
- TOP VIDEO - TOP COMPUTER - Via S. Anna dei Lombardi, 12

Provincia di Napoli
- ITALIANA SOFTWARE - Via Zara, 11 - Acerra
- NUOVA INFORMATICA SHOP - Via Libertà, 185 - Portici

Provincia di Salerno
- COMPUTER WORLD - Via Quarto, 6 - San Marzano sul Sarno

Calabria

Cosenza
- SIRANGELO COMPUTER - Via N. Parisio, 25

Reggio Calabria
- CONTROL SYSTEM - Via S. Francesco da Paola, 49/D-E
- SYSTEM HOUSE - Via Fiume ang. Palestrino, 1

Provincia di Reggio Calabria
- COMPUTER SHOP - Via Matteotti, 48/50 - Locri

Sicilia

Provincia di Enna
- ITALSOFT - Via Dottor Palazzolo - Agira

Palermo
- C.H.P. - Via Libertà, 95

cambia in Commodore®

Esigi sempre la garanzia della Commodore Italiana S.p.A.

Elenco aggiornato al 3 Aprile 1989

LA VOCE DEI LETTORI

VIVAMIGA, ABBASSO MS-DOS

(Gabriele De Pasquali - Savona)

Mi pare a dir poco curiosa la posizione di un vostro collaboratore che ha definito "fantastico" il mondo Ms-Dos (C.C.C. N. 49, pagina 27). Infatti in questo *mondo* il colore dominante è praticamente il verde o il grigio, il suono è uno strepitoso *beep*, la grafica è proprio indimenticabile, i giochi sono ridotti agli scacchi o simili, le animazioni animali rari (a parte il lampeggiare del cursore), il colloquio con la macchina fatto di sequenze di tasti, noioso e pesante da digerire, programmi originali che costano un occhio (e anche le copie non scherzano), mouse praticamente di figura, joystick assente del tutto.

Signori, la mia opinione è che un Ms-Dos va bene per un titolare d'azienda, o simili, che abbia a che fare con contabilità, fatturazione e divertimenti del genere. Io personalmente ho quarant'anni e considero il mio "poco più di un giocattolo" (vedi stesso numero di C.C.C. pagina 10) Amiga 500 un frutto tecnologico che ha dello straordinario e quel che è bello mi serve per tutto: svago e applicazioni serie, e con il massimo della semplicità. Ma se poi qualcuno vuole dare ancora una sbirciatina al mondo Ms-Dos basta il Turbotransformer il quale è in grado di far girare, Commodore dice ed esperienza conferma, parecchi programmi, e con certezza i più famosi, quali il DB III, il Lotus 1-2-3, Wordstar, Multiplan, e altri (a questo riguardo, da prove fatte, il mio Transformer legge e scrive su entrambe le facciate del dischetto, per un totale di 720 K, contrariamente a quanto scritto nell'articolo a pagina 26, stesso fascicolo).

Riferendomi sempre all'articolo di cui sopra, trovo poco serio far risaltare l'enorme differenza nella esecuzione del benchmark, considerato che l'Amiga deve mettersi nei panni di un altro computer e poi svolgere il lavoro (certe cose ingannano certamente i meno esperti). In casi come questo non pensate che sia doveroso pubblicare anche i tempi di esecuzione dell'Amiga, quando lavora come tale, e non in emulazione?

In conclusione mi chiedo perchè volete far piangere un sacco di gente, titolari d'azienda esclusi.

Sperando che questa mia lettera venga aperta da un *Amigo* (altrimenti immagino già il cestino) vi saluto cordialmente.

da compilare non solo "sfusi" ma (mi pare dietro modico compenso) anche su moduli continui, non ci risulta che il *Totocalcio* offra lo stesso servizio.

Un programma di stampa schedine può essere efficiente, e sufficientemente "universale", solo nel caso di utilizzo di moduli continui che, escludendo le prime due schedine di "prova", potrebbe stampare tutte le altre con la massima precisione.

Vi sono, infatti, numerose difficoltà per inserire, e stampare, una schedina alla volta, dal momento che le stampanti esistenti differiscono notevolmente tra loro per ciò che riguarda dimensioni e posizionamento del foglio di carta. Inoltre, nell'inserire ciascuna schedina, bisognerebbe sistemarla con estrema precisione prima di impartire il comando di stampa. In caso contrario la ricevitoria non potrebbe accettare moduli mal compilati.

Questo è l'unico motivo che ci ha impedito di scrivere un programma idoneo.

DISCONTINUITA'

☐ Ho scritto un programma per tracciare grafici di funzioni ma, in caso di discontinuità, l'elaborazione procede in modo indesiderato anche se attivo il comando *On error Goto*. Come posso fare?

(Paolo Trotta - Salerno)

• Sicuramente, all'interno del programma, sono presenti alcune linee che sono necessariamente "universali" nel senso che trasformano il dato numerico nella corrispondente coordinata.

Invece di attivare il comando *On Error Goto* (che, tuttavia, è sempre bene inserire per evitare interruzioni indesiderate) è molto meglio introdurre particolari *filtri software* che, evitando elaborazioni impossibili (come la divisione per zero o il plottaggio di punti esterni al range disponibile) si limitano ad ignorare il dato elaborato.

Ad esempio, una riga del tipo...

```
500 Line (MX , MY/H) - (MX , MY/(H*2))
```

...è opportuno scriverla così:

```
500 If H<>0 Then Line (MX , MY/H) -
(MX , MY/(H*2))
```

In casi come questi, ovviamente, la linea non verrà tracciata; del resto non sarebbe possibile tracciarla nemmeno con carta e penna.

Per avere una risposta più precisa in merito, tuttavia, sarebbe opportuno l'invio del programma in questione (su *disco*).

PUBBLICATEMI

□ Vi ringrazio per aver pubblicato il mio listato per C/16 (vedi C.C.C. N. 62); vorrei sapere se mi spetta un compenso ed il motivo per il quale non avete pubblicato il mio indirizzo.
Claudio Verniani
Via Fossolo, 10
40138 Bologna

• Di solito non pubblichiamo gli indirizzi dei nostri lettori, salvo esplicita sollecitazione (come in questo caso). Spero che, grazie alla pubblicazione del tuo indirizzo, tu possa entrare in contatto con altri utenti di C/16 e Plus/4 in modo da incrementare la tua *mailing list* che vanta nominativi perfino inglesi ed americani.

Per ciò che riguarda il compenso, purtroppo, non hai inviato l'articolo esplicativo su disco (come richiesto diverse migliaia di volte) ma su carta; questo, in ogni caso, era piuttosto simile a quello di L. Emilitri. Il tuo listato è quindi apparso sulle nostre pagine come un surplus di notizie nella speranza che potesse interessare i nostri lettori.

VELOCITA' E SPROTEZIONE

□ Vorrei che parlaste più spesso dei vari velocizzatori e, soprattutto, che spiegaste in che modo uno *Speed Dos*, grazie all'aumento di velocità, può copiare dischi superprotetti come *Geos*.
(Alessandro Zarrilli - Poggibonsi)

• Per quanto riguarda i vari velocizzatori in commercio abbiamo avuto modo di parlarne più di una volta. Ad un collaboratore ho chiesto da tempo la stesura di un articolo ad hoc che, però, tarda ad esser partorito.

Posso solo dire, per ora, che la velocità di caricamento e/o salvataggio di dati e programmi non ha nulla a che fare con le sprotezioni. Semmai la domanda è mal formulata e non certo per colpa del nostro lettore.

Di solito, infatti, qualsiasi cartuccia in commercio vanta diverse caratteristiche tra cui, quasi sempre, l'aumento di velocità e la possibilità di copia di programmi protetti. Gli utenti, quindi, pensano che il primo vantaggio sia da correlare con il secondo, e viceversa, ma non sempre è così.

Il fatto è che ormai per il C/64 hanno già inventato tutto (e qualcosa in più). Per convincere il probabile cliente all'acquisto di una nuova cartuccia i progettisti sono "costretti" ad inserire, nelle Rom, anche le features non strettamente necessarie, tra cui la copia di dischi (e nastri) superprotetti.

E' ovvio che le cartridge più recenti hanno una maggiore probabilità di forzare sprotezioni di nuova concenzione eventualmente escogitate.

Di solito, però, capita molto più frequentemente di copiare dischi già piratati, piuttosto che quelli originali (e chi non è d'accordo scagli il primo dischetto). In questi casi non sono presenti protezioni da rimuovere, ma è sufficiente un copiatore, magari veloce, in grado di riprodurre fedelmente un dischetto apparentemente protetto.

JANUS AT; FUNZIONA?

□ Vorrei sapere se la scheda *Janus* per Amiga (che trasforma questo computer in un Ms-Dos AT) è realmente compatibile oppure presenta inconvenienti, come mi sembra di capire da varie voci che circolano.
(Cosimo Scialla - Eboli)

• Siamo contrari, per principio, alle voci che circolano (la calunnia è un venticello facile a sorgere e diffondersi, difficile da bloccare).

In precedenza, più di una volta, abbiamo espresso le nostre perplessità su una scheda che, a pensarci bene, costa quasi come un vero computer Ms-Dos. Tuttavia, sempre per principio, sarebbe opportuno che chi mette in giro le "voci" (di cui molti rimangono vittime, in perfetta buona fede, tra cui il nostro affezionato lettore) dica *esattamente* con quali programmi o in quali situazioni la scheda non svolge il suo lavoro.

In caso contrario, taccia per sempre.

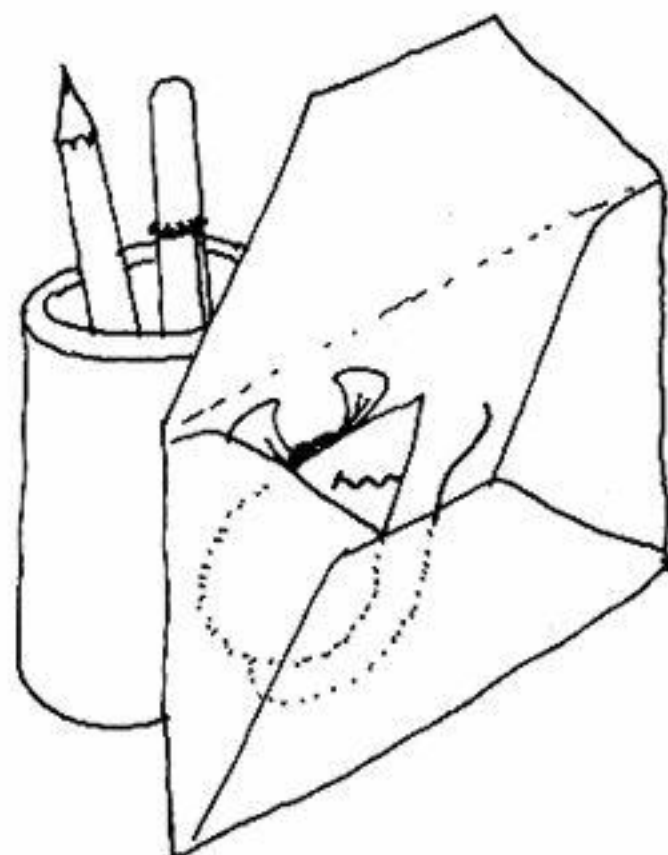
BUONI CONSIGLI

□ So benissimo che non date consigli sulle periferiche da acquistare, ma vorrei un'indicazione su hardware e software per usare il mio C/64 nella stesura di testi, di cartoncini di auguri e simili.
(Dario Cardile - Messina)

• Il programma che prediligo è Easy Script perchè è piuttosto breve (: si carica in fretta), potente e versatile. In poche parole, soddisfa l'utente anche più esigente. Per realizzare cartoncini di auguri, disegnini e carte intestate, uso da tempo altri computer dotati di programmi DTP che attivano stampanti laser. Ricordo, però, che quando usavo il C/64 per questi scopi trovavo molto divertente *Print Shop* e *Print Master*. So che vi sono in giro molti altri programmi che, se compatibili con lo Speed Dos, rendono la procedura sufficientemente spedita e sicura.

Rimane da parlare della stampante, sulla quale non mi pronuncio per una serie di motivi più volte accennati.

Posso solo dire che la Mps-803, per fortuna, è uscita di produzione; nonostante fosse molto affidabile era infatti piuttosto lenta e poco "professionale".

Rimane il solito consiglio: anzitutto è *indispensabile* una stampante MPS-803 compatibile (pena il mancato funzionamento di *tutti* i programmi grafici); in secondo luogo è bene privilegiare il negoziante che vi consente di effettuare prove (*prima* dell'acquisto) sull'eventuale modello *non* Commodore che vi propone. Le stampanti Commodore specifiche per il C/64 (ed è quasi inutile ricordarlo) sono, infatti, realmente e totalmente 803 compatibili.

risposte rapide

SOLO SU DISCO

(Andrea Andenna)
(Luca Cassioli - S. Angelo)
(Vincenzo Mai - Avezzano)

Posso prendere in considerazione solo listati, corredati da articoli, inviati su *disco*. Inoltre sono privilegiati i programmi dettagliatamente spiegati e, se in linguaggio macchina, solo se corredati dal disassemblato *commentato*.

Per quanto riguarda eventuali malfunzionamenti di programmi digitati da rivista, successivamente vengono indicati, se effettivamente presenti, i bug riscontrati. In caso contrario è necessario inviare, su *disco*, il programma che presenta inconvenienti per consentirci di individuare gli errori.

PARLIAMONE

(Giacomo Salmeri - Milano)

La proposta di collaborazione sembra interessante. Aspetto una telefonata per stabilire un accordo.

PER ESPERTI

(Franco Molinaro - Imperia)
(Ivan Scillone - Udine)

Il sistema operativo Ms-Dos ed il linguaggio interprete Gw-Basic richiedono interi volumi di spiegazioni. Il nostro emulatore per C/64 (che ha raggiunto la terza ristampa) si rivolge quindi ad utenti che, in un modo o un altro, siano già in grado di usare correttamente la sintassi richiesta.

BUG IN EMULATORE GW-BASIC

(Osvaldo Graziano - Amantea)

Sul N. 52 ("Scrivere in Ms-Dos") sono riportati tre file da usare in ambiente Ms-Dos Emulator, tra cui *Print1* che elimina il bug di scrittura dei file lamentato.

AVVISO AI COLLABORATORI

Gli articoli ed i programmi che appaiono sulle pagine della nostra rivista vengono *sempre* compensati, tranne casi particolarissimi (come la rubrica "La vostra posta"). L'invio dell'assegno, la cui entità è fissata a nostro insindacabile giudizio, viene effettuato tra il 40mo ed il 70mo giorno a partire dalla data di pubblicazione dell'articolo stesso (data riportata sul fascicolo di C.C.C.). A volte, purtroppo, capita che gli assegni vengano smarriti nel tradizionale marasma postale. Coloro che, pertanto, non ricevessero il compenso che si aspettano (soprattutto se, in precedenza, sono stati contattati a tale scopo dal nostro ufficio amministrativo) sono pregati di informarci del mancato ricevimento dell'assegno non oltre il 65mo giorno dalla pubblicazione. In caso contrario non avranno seguito eventuali proteste a causa dell'impossibilità di effettuare ricerche presso gli uffici postali.

METTI UN TURBO NELL'AMIGA

La scheda A-2620, per Amiga 2000, non è una semplice espansione di memoria; nè la sua forza è dovuta, soltanto, al 68020

di **Stefano Simonelli**

Amiga è uno tra i personal computer più potenti attualmente disponibili sul mercato europeo; la versione 2000 è sicuramente la più versatile, grazie all'elevato numero di slot di espansione che le permettono di aumentare le capacità operative. E' infatti disponibile una scheda acceleratrice (siglata A-2620) che mette un turbo, nel vero senso della parola, al vostro beniamino.

Il nuovo, potente "motore" si concretizza in una scheda su cui primeggia la CPU 68020 affiancata dall'unità a virgola mobile 68881 (in grado di eseguire operazioni matematiche rapidissime); sul circuito stampato è altresì presente un'espansione Ram da 2 megabytes a 32 bit (con slot per l'ulteriore espansione a 4 MByte) e 64 K di Rom (espandibili a 128 K) impiegati per diagnosi, opzioni di boot-up e particolari inizializzazioni usate da configurazioni software alternative (come, ad esempio, il sistema operativo UNIX).

E' bene precisare che l'aggiunta del microprocessore 68020 non rende più veloci gli accessi al drive; tuttavia, utilizzando la memoria presente nella scheda A-2620, è possibile attivare un programma di nome FACCII (Floppy Accelerator) che permette di sveltire notevolmente le operazioni in cui sono coinvolte le procedure di accesso al disco.

Il funzionamento del FACII è simile al noto comando ADDBUFFERS del CLI, vale a dire che riserva uno spazio di memoria nel quale vengono memorizzati i files letti dal drive.

Per meglio chiarire il concetto ricorreremo ad un esempio.

Quando, da Cli, impartiamo il comando DIR, l'AmigaDos carica dapprima il comando in memoria; in seguito lo esegue leggendo i settori nei quali è contenuta la directory del disco, li elabora secondo alcuni parametri e visualizza il risultato sul video.

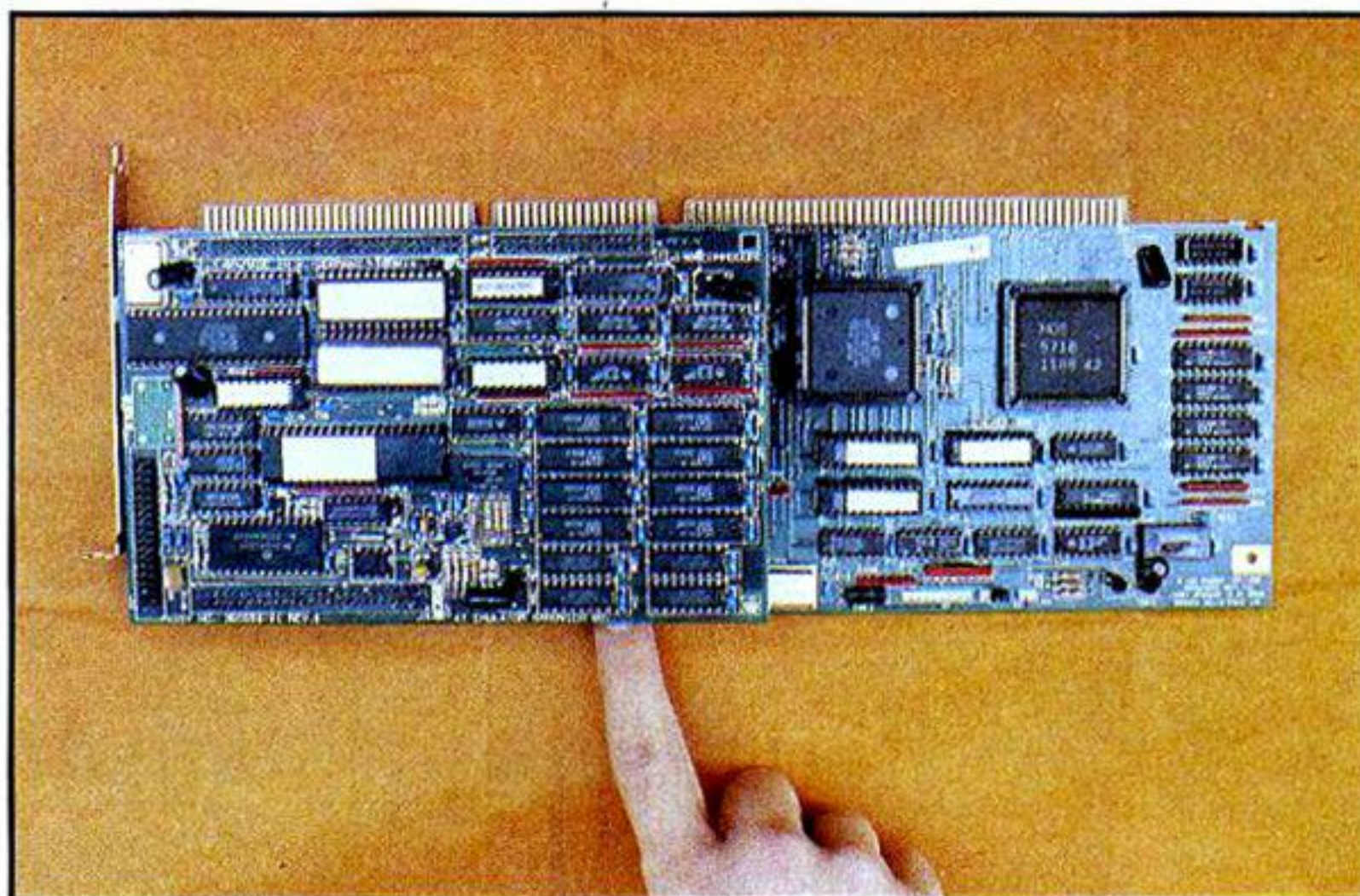
Desiderando, subito dopo, rivedere la stessa directory, Amiga è costretta a ripetere, uno per uno, i passi che ha appena eseguito, con notevole spreco di tempo. Ricorrendo, invece, al FACCII, il comando DIR e la directory vengono immagazzinati in Ram in appositi buffer in modo tale che la seconda volta che se ne chiede la visualizzazione, la directory appare istantanea-

mente, eliminando le lunghe attese.

Le operazioni descritte sono svolte, in verità, anche dal comando CLI Addbuffers, con la differenza che quest'ultimo può utilizzare solo memoria CHIP e quindi non è in grado di sfruttare l'eventuale memoria aggiuntiva presente. Inoltre la dimensione del buffer non è variabile: una volta settata non può essere diminuita, ma solo aumentata fino ad esaurimento della memoria centrale.

SEMPRE PIU' IN ALTO

Il programma FACCII possiede un'altra caratteristica strabiliante. Supponiamo, per esempio, che in un certo momento, nel sistema, siano liberi solo 300 K Ram e che FACII occupi, nello stesso istante, 150 K Ram; volendo caricare in memoria un'applicazione che richiede l'utilizzo di 400 K, FACCII "capisce" la situazione e libera la memoria necessaria per far girare il programma in questione. Ciò, e qui è il bello, non avviene in modo casuale: FACCII è in grado di distinguere i dati dalle informazioni nelle directory; perciò cercherà di mantenere in memoria le informazioni mettendo a disposizione la memoria occupata dai dati.

E' da notare il fatto che FACCII, per svolgere le sue funzioni, utilizza la memoria e non il microprocessore. Ne consegue che è possibile ottenere gli stessi vantaggi procurandosi solo il programma FACCII (e memoria Ram aggiuntiva), continuando ad operare con il "vecchio" 68000. La procedura, però, risulterà più lenta poichè lo stesso programma non verrà elaborato alla maggior velocità tipica del 68020, micro montato sulla potente scheda aggiuntiva.

VELOCITA'

La velocità con cui opera la scheda è approssimativamente di 14.3 MHz, cioè il doppio della velocità di clock del 68000 dell'Amiga. Ciò significa che si riesce ad ottenere un aumento delle prestazioni fino al 400% con programmi che utilizzano prevalentemente la CPU. Con i programmi, infatti, che sfruttano soprattutto il FAT AGNUS, il BLITTER o il COPPER, non si riscontrerà un elevato aumento delle prestazioni, poichè quando il 68020 accede alle risorse interne di Amiga, rallenta la sua velocità dal momento che i coprocessori interni lavorano solo a 7.2 MHz.

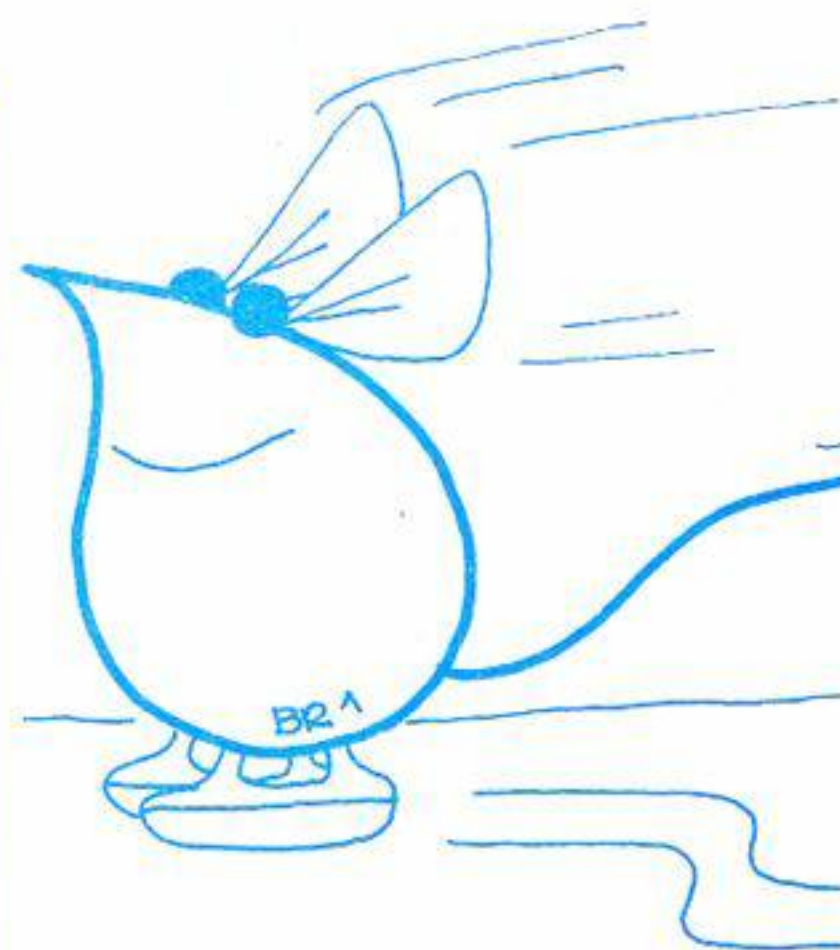
Per ciò che riguarda il coprocessore matematico 68881 è bene precisare che esso entra (automaticamente) in funzione solo se il software prevede la sua presenza. Ne consegue che se un'applicazione non è stata concepita in modo da sfruttare il coprocessore matematico in virgola mobile, averne uno nel sistema non serve assolutamente a nulla. La CPU, infatti, non si accorge di avere a disposizione un circuito integrato studiato appositamente per applicazioni matematiche, e continua ad elaborare, come sua abitudine, tutte le routine in virgola mobile.

Attualmente soltanto pochi programmi sfruttano le capacità di calcolo del coprocessore matematico. Risulta quindi difficile esprimere un giudizio obiettivo sull'utilità e l'efficenza di una scheda comprendente un 68881.

Abbiamo sottoposto il nostro A-2000 a numerose prove di velocità, come si può notare dalla tabella, soprattutto con programmi che richiedevano enormi quantità di calcoli. Si è riscontrato che vi è un notevole aumento di velocità soprattutto utilizzando il BASIC; perfino l'editor di Abasic (che normalmente è molto lento ed innervosisce anche gli utenti più calmi) diventa veloce e piacevole da utilizzare.

Ritornando al 68020, è da ricordare che questo microprocessore è superiore al 68000 non solo per la velocità di esecuzione, ma soprattutto perchè è supportato da un maggior numero di istruzioni Assebly. Inoltre è capace di gestire la memoria virtuale, vale a dire che può eseguire programmi in linguaggio macchina che non risiedono in memoria centrale ma che

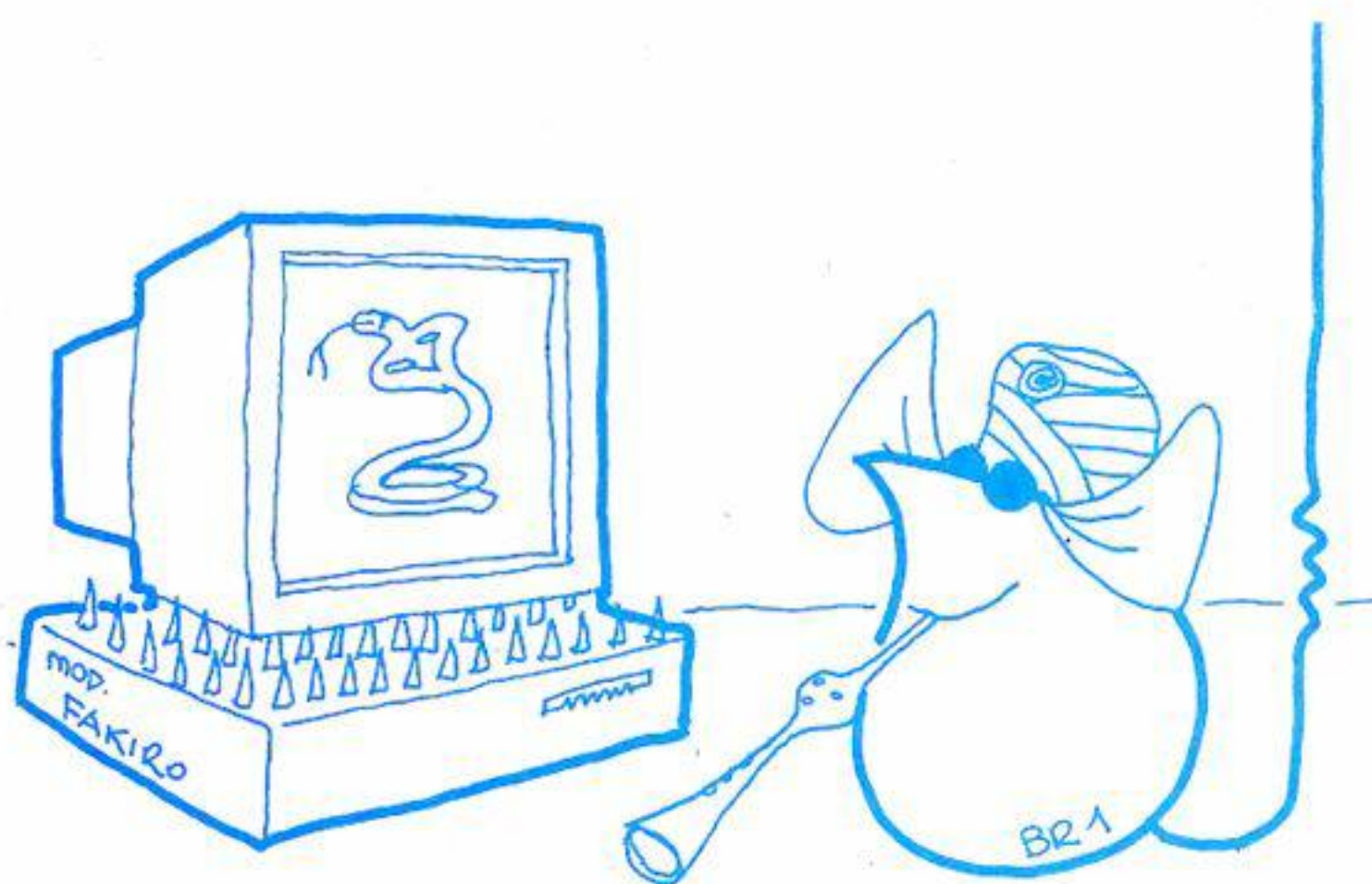
Test	Microprocessore 68020	68000
	Tempi rilevati (min. sec. dec.)	
for i=0 to 10000 a = log(sin(abs(rnd(i)))) b = (i+1) c = a / b next i	1.10.67	3.78.82
fill page photon paint	0.18.17	0.24.01
wrapon cone photon paint	0.45.11	0.57.38
mandelbrot: il dragone	11.33.45	13.20.12

(N.B: wrapon cone = arrotola una videata intorno ad una figura solida di forma conica.)

sono presenti sulla memoria di massa (:disco). In tal modo, senza difficoltà, si possono creare, e far eseguire, programmi che non risiedono interamente in memoria, ma che si autocaricano un pezzo alla volta.

ALTRE CONSIDERAZIONI

La compatibilità con programmi già esistenti è molto elevata; è stata comunque prevista la possibilità di disabilitare il 68020 standard. Resettando A-2000 (tenendo premuti ambedue i tasti del mouse) viene visualizzato un menu che permette di settare il computer in modalità UNIX, 68020 oppure 68000 "normale". In quest'ultimo caso la macchina si presenta come se non fosse presente la scheda A-2620 e non rende disponibli neppure i 2 MByte di Ram; il tutto, naturalmente, per una totale compatibilità con i programmi in commercio.

Per ciò che riguarda l'installazione delle scheda, non sono stati riscontrati problemi di sorta, grazie soprattutto al manuale, molto chiaro ed esplicito al riguardo; viene infatti descritto, passo per passo, come effettuare l'inserimento, operazione che chiunque, pur se digiuno di esperienza, può effettuare.

L'unico lato negativo della scheda è il suo... prezzo, quasi superiore a quello dello stesso computer. A poco serve consolarsi pensando al fatto che il solo costo di una semplice espansione di 2 MByte di Ram (ma a 16 bit) è superiore al milione li lire. La scheda A-2620 è prezzata con 2.700.000 lire, nelle quali sono compresi i 2 MByte di Ram (da ben 32 bit). Tutto sommato è forse più conveniente, fatte le debite proporzioni, dotare il proprio Amiga 2000 di una scheda 68020 piuttosto che di una semplice espansione Ram.

la Repubblica

Direttore Eugenio Scalfari

MANO AL PORTFOLIO

Affidiamo al nostro Commodore la gestione di uno dei più diffusi giochi a premi quotidiani; e chisssà che non ci porti fortuna...

di **Domenico Pavone**

Da un paio di anni, ormai, una nuova consuetudine è andata sempre più radicandosi tra i lettori del quotidiano La Repubblica: quella di controllare (non necessariamente dopo aver letto il giornale), calcolatrice alla mano, se la spesa di acquisto non si è trasformata in un più remunerativo investimento economico.

Ci riferiamo, ovviamente, alla particolare lotteria, Portfolio Vip, ben conosciuta anche dai non affezionati lettori della testata in questione.

Chi, infatti, non ha un parente, un amico o un semplice conoscente, che non custodisce, più o meno segretamente un cospicuo malloppo di tesserine telate?
Sarebbe il caso, sempre che non possieda anche lui un computer, proporre, al personaggio di cui sopra, di occuparcene noi, magari dietro misero compenso (che ne dite del 60% della eventuale vincita?).

Se state pensando ad ipotetici intrallazzi tra "Repubblica" e "Commodore Computer Club", ebbene siete fuori strada.

Il fatto è che, per le caratteristiche algebrico-aritmetiche, il gioco Portfolio si presta particolarmente ad essere "trattato" da un computer, e non con motivazioni fini a se stesse.

Ottimizzando una procedura "ad hoc", infatti, il risultato più tangibile sarà un enorme risparmio di tempo nel controllo dei vari elementi del gioco; per non parlare della maggiore attendibilità dei risultati.

Il tutto, tra l'altro, con il minimo sforzo.

A dispetto, infatti, dei megaprogrammi già apparsi un po' dappertutto, bastano poche righe di Basic per ottenere una valida e completa routine che consenta di memorizzare un numero illimitato di tessere, nonchè di aggiungerne, nel tempo, un altrettanto illimitato numero.

La stessa, ovviamente, dovrà occuparsi del controllo quotidiano di eventuali vincite, riducendo al minimo gli input necessari.

Quest'ultima caratteristica, come vedremo meglio tra breve, rende il programma di comoda fruizione anche per chi possiede solo una o due tesserine, in quanto non richiede che vengano immesse le 48 "variazioni" pubblicate quotidianamente.

Ma bando alle ciance, e veniamo al sodo.

ISTRUZIONI PER L'USO

Il listato di queste pagine è scritto in un Basic che più Basic non si può, totalmente privo di Poke, in modo da renderlo integralmente compatibile con i vari C/64, Plus-4 o C/128, purchè dotati di disk drive.

Anche gli utenti di Amiga, con minime e banali modifiche, potranno comunque approfittare dell'occasione, riferendosi al riquadro loro dedicato.

E' chiaro che, in funzione della (quasi) totale trasportabilità, l'estetica dell'output sullo schermo è ridotta all'essenziale.

D'altra parte nulla vieta (anzi è auspicabile) un vostro intervento in questa direzione; rendere meno scarna la rappresentazione grafica del programma non è poi un'impresa così ardua, soprattutto se si può approfittare delle varie routine dell'enciclopedia Basic, da noi pubblicate nel corso degli anni.

Si copi, dunque, il listato (le Rem possono essere tralasciate), lo si salvi su disco, e si impartisca Run.

Apparirà un menu con quattro opzioni, di immediata comprensione, selezionabili digitandone il relativo numero.

La prima volta che si utilizza il programma occorrerà cominciare con la scelta 1, che chiederà, anzitutto, quante tessere vogliamo memorizzare.

Se ne digiti il numero, quindi si prema Return.

Inizierà ora la fase di input degli otto "titoli" di ogni tessera, che verrà ripetuta per il numero totale delle stesse.

E' conveniente, in questa fase, annotare sulla tesserina quello che diventerà il suo numero di identificazione, per poterla eventualmente rintracciare con facilità, e soprattutto in funzione (lo vedremo tra breve) delle eventuali future aggiunte.

Quindi, se per esempio appare la richiesta "Tessera 3 - Titolo N. 1", scrivete con un pennarello (o adoperando targhette autoadesive) il numero 3 sulla tesserina.

A conclusione delle operazioni riguardanti ogni singola tessera, verrà chiesta conferma sulla validità dei dati: se non si sono commessi errori si prema Return; in caso contrario, la pressione di qualunque altro tasto riproporrà gli input della stessa tessera, consentendone la reimmissione.

Ultimata questa fase, il programma provvederà a memorizzare sul disco presente nel drive un file User di nome "portfol.tessere", contenente sia il numero di tessere che i loro dati.

Qualora, successivamente, si intendesse aggiungerne altre, basterà caricare il programma e, dopo averlo mandato in esecuzione, scegliere l'opzione 2.

Si badi che, stavolta, nel disco in opera dovrà già essere presente il file "portfol.tessere", cui la routine accederà come prima operazione.

Comunque, a scanso di involontari "pasticci", tutti gli accessi al disco sono forniti di controllo di errore che, in caso di ano-

ANCHE PER AMIGA

Differenze sintattiche a parte, lo stesso algoritmo seguito per i "piccoli" della Commodore, può essere facilmente tradotto per il ben più potente elaboratore a sedici bit.

Occorrre, però, tenere presente alcune differenze di comportamento, che richiedono qualche procedura più accurata, rilevabile nel listato riservato agli utenti di Amiga.

Precisiamo, prima di ogni altra cosa, che il listato mantiene alcuni numeri di riga (peraltro consentiti), senza ricorrere a più chiare Label, per adeguarsi alle note descrittive presenti nell'articolo, che pertanto risultano valide anche per la versione Amiga.

Per lo stesso motivo non è stata curata a fondo la classica rappresentazione estetica dei listati di Amiga, lasciando a voi il compito di provvedere alle varie indentazioni (sempre che ne abbiate voglia).

L'unico aspetto degno di nota è rappresentato dalle righe che seguono l'eventuale scelta 1 dal menu principale.

Mentre, infatti, il Dos delle altre macchine segnala un "File exists" se si tenta di utilizzare l'opzione 1 quando il file "portfol.tessere" è già stato creato una prima volta, l'Amiga Dos... se ne impipa.

Nel senso che, come è noto, si limiterebbe a riscriverlo, con conseguente perdita dei dati precedenti.

Per evitare ciò, si ricorre ad un machiavellico inganno ai danni del Dos, basato sull'istruzione On Error.

Quando si sceglie l'opzione 1, viene aperto in input il file "portfol.tessere": se non è presente, verrà generata una segnalazione di errore (err=53, cioè file not found) che, intercettata dalla riga 1000, consentirà la prosecuzione del programma.

Se invece il file viene effettivamente aperto (e quindi è già presente), la variabile di sistema Err varrà 0 (=nessun errore), e la routine si interromperà intimando di adoperare l'opzione 2.

Per gli stessi motivi, nel caso di nuovi inserimenti nell'archivio tessere, il programma non provvede a cancellare il file precedente: è sufficiente riaprirlo in output.

La segnalazione di eventuali altri tipi di errori (File not found, e così via) è stata lasciata all'interprete.

Quindi, se per esempio non è presente sul disco il file-dati e si sceglie l'opzione 2, si avrà un arresto del programma con relativo (odiosamente lento) riquadro che demarca l'errore sul listato.

Salvo, ovviamente, vostri auspicabili interventi.

Si tenga anche presente che, a differenza di quanto descritto nell'articolo a proposito degli input da tastiera, l'interprete di Amiga Basic non mantiene le precedenti assegnazioni (nelle reimmissioni dovute ad errori) premendo il solo Return in fase di input.

malie, bloccano il programma segnalando il tipo di problema.

Tornando alle nostre aggiunte, verranno ripetuti i medesimi input visti per la fase 1, ma con numero di tessera aggiornato alla nuova situazione, da annotare sulla stessa.

Se, per esempio, con l'opzione 1 si erano archiviati i dati di 5 tessere, al momento di aggiungerne un'altra con l'opzione 2, il primo messaggio di input risulterà: "tessera 6 - titolo n.1".

In pratica, prima di immettere i dati, bisognerà annotare un bel "6" sulla nuova tessera della nostra collezione.

GIOE O DOLORI?

E veniamo all'ultima opzione, quella che deciderà sulla nostra sorte di eventuali neo-milionari (ma non troppo).

Attivando l'opzione 3 del menu, infatti, si passa al controllo delle tessere di cui disponiamo, per verificare se una (o più) di loro raggiunge o supera il famigerato "dividendo" pubblicato ogni giorno assieme alle 48 variazioni di borsa.

In pratica, dopo avere trasferito in memoria i dati del file "portfol.tessere", il programma richiede in input tutti i numeri dei titoli, ma, per farci risparmiare fatica, escludendo quelli che non sono presenti in nessuna delle nostre tessere.

La cosa può risultare molto utile per chi ne possiede un numero limitato: con due soltanto, per esempio, al massimo occorrerà immettere 16 valori, compito ben più confortevole che digitare, per intero, i 48 titoli di borsa.

Se però, come molti, si dispone di oltre dieci tessere, quasi certamente il numero di input si avvicinerà ai fatidici 48.

In questo caso, talvolta può risultare scomodo controllare visivamente quale numero prendere in considerazione e quale no.

Volendo digitare in sequenza i 48 titoli, basterà comunque eliminare dal listato il riferimento alla variabile FL(t(z,y)) in riga 320 e tutto ciò che è presente in riga 490, da IF in poi.

Il programma, anche qui, chiede conferma sull'esattezza dei valori immessi, in caso positivo richiede in input il dividendo da considerare vincente, e passa ad eseguire tutti i calcoli necessari, tessera per tessera.

Se il totale algebrico dei titoli compresi in una tessera è uguale al dividendo vincente, segnala la vincita del premio Portfoliò, se maggiore (beati voi!) quella del premio Vip.

Nel caso si sia immesso un dato errato, si tenga presente che non è indispensabile fornire di nuovo tutti gli input, purchè ci si ricordi (:si annoti) il numero del titolo non corretto.

Quando infatti, dopo la richiesta di conferma, si torna al primo input, digitando solo Return viene conservata l'assegnazione precedente: basterà dunque premere Return per tutti gli input, tranne quelli da correggere.

Per i semplici "users" non c'è altro da aggiungere, mentre, per chi non si accontenta dell'usa e getta, l'argomento non è ancora chiuso.

QUALCHE NOTA SUL LISTATO

Lo sviluppo del programma, redatto secondo i più noti dettami del Basic 2.0, non presenta particolari difficoltà, se non quella di seguire il flusso della routine, direzionato dal valore della variabile A, stabilito all'atto della scelta nel menu (righe 190-220).

Per la fase di memorizzazione delle tessere, gli input da tastiera sono introdotti nella matrice a due indici T(z,y), con Z che fa riferimento al numero della tessera, mentre X rappresenta uno degli otto titoli all'interno della stessa.

Va da sè che, in fase di dimensionamento (Dim), Y avrà come valore massimo 8, mentre Z verrà dedotto dal primo input, che richiede, appunto, il numero di tessere da memorizzare.

L'array viene poi archiviato sequenzialmente sul file user opportunamente aperto col nome "portfol.tessere" (420-450), preceduto però dalla variabile xx, cui corrisponde il numero di tessere.

Adottando tale sistema, diventa poi facile l'eventuale inserimento di nuove tessere.

Basterà infatti:

- Prelevare dal file il primo dato, cioè il numero di tessere già archiviate, e dimensionare con esso la matrice che conterrà tutti i dati delle tessere.

- Caricare in memoria, con opportuni cicli For...Next, l'intero array memorizzato nel file user.

- Accodare, sempre alla stessa matrice già in memoria, i nuovi dati inseriti da tastiera, aggiornando il numero totale delle tessere.

- Cancellare il precedente file user dal dischetto, e riscriverlo inserendovi il nuovo array (ora più lungo), sempre preceduto dal numero delle tessere.

La tecnica adottata, che forse non brilla per eleganza, potrebbe essere sostituita dalle possibili funzioni Append legate ai file sequenziali.

Tuttavia, non dovendo trattare quantità esorbitanti di dati, si dimostra piuttosto veloce ed efficace, soprattutto considerandone la maggiore concisione in fase di stesura (per voi = copiatura) del programma.

Poco da dire sull'opzione di controllo sulle vincite, dato che, una volta caricato il solito array in memoria, si limita ad eseguire, per ogni tessera, una semplice somma algebrica delle "variazioni" corrispondenti ai titoli in essa contenuti.

Unico punto degno di nota, la variabile FL(I), che funge da "flag" sugli input da tastiera delle variazioni giornaliere, come può essere meglio chiarito con un esempio.

Tutti gli elementi del vettore FL(I), dimensionato con indice massimo 48, inizialmente varranno 0.

Supponiamo di avere in memoria (dopo il prelievo da disco) solo una tessera, che contenga i titoli 2 e 10 (in realtà saranno 8, ma semplifichiamo per brevità).

Ebbene, se si assegna un valore diverso da zero (nel nostro caso, 1) agli elmenti FL(2) e FL(10), eviteremo di richiedere in input tutte le variarioni giornaliere, limitandoci ad impostare solo quelle il cui flag corrispondente è diverso da zero (riga 490).

Quindi, tornando all'esempio citato, gli unici input necessari riguarderanno le variazioni numero 2 e 10, mentre le altre 46 verranno ignorate.

Questa piccola accortezza rende "ergonomica" la nostra routine, dato che in sua assenza, e possedendo solo qualche tessera, risulterebbe più pratico eseguire i calcoli con una normale calcolatrice tascabile, piuttosto che perdere tempo a digitare i 48 numeri necessari per controllare la vincita.

E, in questo caso più che mai, il tempo è denaro...

```
100 REM ************************************************************
110 REM         GIOCO "LA REPUBBLICA - PORTFOLIO (VIP)"
120 REM  PER C16, C64, PLUS4 E C128 (MODO 128) + DISK DRIVE
130 REM ************************************************************
140 :
150 CL$=CHR$(147):PRINTCL$: B$="PREMI RETURN SE TUTTO OK"
160 PRINT"1 - PRIMA MEMORIZZAZIONE TESSERE"
170 PRINT"2 - AGGIUNTA NUOVE TESSERE"
180 PRINT"3 - CONTROLLO VINCITA":PRINT"4 - FINE":OPEN15,8,15
190 GETA$:A=VAL(A$):IFA$=""THEN190
200 IFA=4THENPRINTCL$:END
210 IFA=1THEN290
220 IFA<>2ANDA<>3 THEN190
230 REM ------------------------------------------------------------
240 REM                 LETTURA DATI DAL DISCHETTO
250 REM ------------------------------------------------------------
260 PRINTCL$"ATTENDI...":OPEN5,8,5,"PORTFOL.TESSERE,U,R"
270 INPUT#15,E,E$:IFE<>0THEN PRINTCL$+E$:CLOSE5:CLOSE15:END
280 INPUT#5,X$:X=VAL(X$):DIMFL(48):IFA=3THEN300
290 PRINTCL$"QUANTE TESSERE";:IFA=2THENPRINT" DA AGGIUNGERE";
295 INPUTXX:IFXX=0THENCLOSE5:CLOSE 15:RUN
300 XX=X+XX:DIM T(XX,8):IFA=1THEN370
310 FORZ=1TOX:FORY=1TO8:INPUT#5,TT$
320 T(Z,Y)=VAL(TT$):FL(T(Z,Y))=1:NEXT:NEXT :CLOSE5
330 IFA=3THENCLOSE15:DIMV(48):PRINTCL$:GOTO 490
340 REM ------------------------------------------------------------
350 REM          INPUT E MEMORIZZAZIONE DATI DELLE TESSERE
360 REM ------------------------------------------------------------
370 PRINTCL$:FORZ=1+X TO XX:FORY=1TO8:PRINT"TESSERA"Z;
380 PRINT"- TITOLO N.";Y;:INPUTT(Z,Y):NEXTY:PRINT:PRINTB$
390 GETOK$:IFOK$=""THEN390
400 IFOK$<>CHR$(13)THENZ=Z-1:PRINT
410 PRINT:NEXTZ:IFA=2THENPRINT#15,"S:PORTFOL.TESSERE"
420 OPEN5,8,5,"PORIFOL.TESSERE,U,W":INPUT#15,E,E$
430 IFE<>0THEN PRINTCL$+E$:CLOSE5:CLOSE15:END
440 PRINT#5,STR$(XX):FORZ=1TOXX:FORY=1TO8
450 PRINT#5,STR$(T(Z,Y)):NEXT:NEXT:CLOSE5:CLOSE15:RUN
460 REM ------------------------------------------------------------
470 REM              CONTROLLO  EVENTUALI  VINCITE
480 REM ------------------------------------------------------------
490 FORI=1TO48:IFFL(I)=0THEN510
500 PRINT"VAPIAZIONE N.";I;:INPUT"= ";V(I)
510 NEXT:PRINT:PRINT B$
520 GETOK$:IFOK$=""THEN520
530 IFOK$<>CHR$(13)THENPRINTCL$:GOTO490
540 PRINT:PRINT:INPUT"DIVIDENDO DI OGGI";DI:PRINT:PRINT
550 PR=0:FORZ=1TOXX:S=0:FORY=1TO8:B=T(Z,Y):S=S+V(B):NEXTY
560 PRINT:PRINT"DIVIDENDO TESSERA N.";Z" = ";S;
570 IFS=DITHENPRINT"PREMIO PORTFOLIO SU TESSERA N.";Z;:PR=1
580 IFS>DITHENPRINT"PREMIO VIP SU TESSERA N.";Z;:PR=1
590 NEXTZ:IFPR=0THENPRINT:PRINT:PRINT"NIENTE DA FARE!":END
600 PRINT:PRINT:PRINT"COMPLIMENTI, CE L'HAI FATTA!":END
```

```
'-----------------------------------------------------------
'          PORTFOLIO VIP * versione per Amiga
'-----------------------------------------------------------
    CLS:b$="PREMI RETURN SE TUTTO OK"
    PRINT "1 - PRIMA MEMORIZZAZIONE TESSERE"
    PRINT "2 - AGGIUNTA NUOVE TESSERE"
    PRINT "3 - CONTROLLO VINCITA":PRINT "4 - FINE"
190 a$=INKEY$:a=VAL(a$):IF a$="" THEN GOTO 190
    IF a=4 THEN CLS:END
    IF a=1 THEN
    ON ERROR GOTO 1000
      OPEN "i",#1,"portfol.tessere"
      IF e=0 THEN
        CLS:PRINT"USA L'OPZIONE 2!!!"
        CLOSE
        END
      END IF
      CLOSE #1
      ON ERROR GOTO 0
      GOTO 290
    END IF
    IF a<>2 AND a<>3 THEN GOTO 190
'-----------------------------------------------------------
    CLS:PRINT "ATTENDI..."
    OPEN "I",#1,"portfol.tessere"
    INPUT #1,x$:x=VAL(x$):DIM f1(48):IF a=3 THEN GOTO 300
290 CLS:PRINT "QUANTE TESSERE";
    IF a=2 THEN PRINT " DA AGGIUNGERE";
    INPUT xx: IF xx=0 THEN CLOSE #1:CLEAR:RUN
300 xx=x+xx: DIM t(xx,8): IF a=1 THEN GOTO 370
    FOR Z=1 TO x: FOR y=1 TO 8:INPUT #1,tt$
    t(Z,y)=VAL(tt$):f1(t(Z,y))=1:NEXT:NEXT:CLOSE #1
    IF a=3 THEN DIM v(48):CLS:GOTO 490
'-----------------------------------------------------------
370 CLS:FOR Z=1+x TO xx:FOR y=1 TO 8:PRINT "TESSERA"Z;
    PRINT "-titolo n.";y;:INPUT t(Z,y):NEXT:PRINT:PRINT b$
390 ok$=INKEY$:IF ok$="" THEN GOTO 390
    IF ok$<>CHR$(13) THEN Z=Z-1:PRINT
    PRINT:NEXT:OPEN "O",#1,"portfol.tessere"
    PRINT #1,STR$(xx):FOR Z=1 TO xx:FOR y=1 TO 8
    PRINT #1,STR$(t(Z,y)):NEXT:NEXT:CLOSE #1:RUN
'-----------------------------------------------------------
490 FOR i=1 TO 48:IF f1(i)=0 THEN GOTO 510
    PRINT "VARIAZIONE N.";i;:INPUT"= ";v(i)
510 NEXT:PRINT :PRINT b$
520 ok$=INKEY$:IF ok$="" THEN GOTO 520
    IF ok$<>CHR$(13) THEN CLS:GOTO 490
    PRINT :PRINT :INPUT"DIVIDENDO DI OGGI";di:PRINT:PRINT
    PR=0:FOR Z=1 TO xx:s=0:FOR y=1 TO 8:b=t(Z,y):s=s+v(b)
    NEXT:PRINT :PRINT "DIVIDENDO TESSERA N.";Z;" = ";s;
    IF s=di THEN PRINT "PREMIO PORTF. SU TESSERA N.";Z;:PR=1
    IF s>di THEN PRINT "PREMIO VIP SU TESSERA N.";Z;:PR=1
    NEXT:IF PR=0 THEN PRINT:PRINT"NIENTE DA FARE!":END
    PRINT:PRINT:PRINT "COMPLIMENTI, CE L'HAI FATTA!":END
'-----------------------------------------------------------
1000 e=ERR
     IF e=53 THEN
     RESUME NEXT
     END IF
```

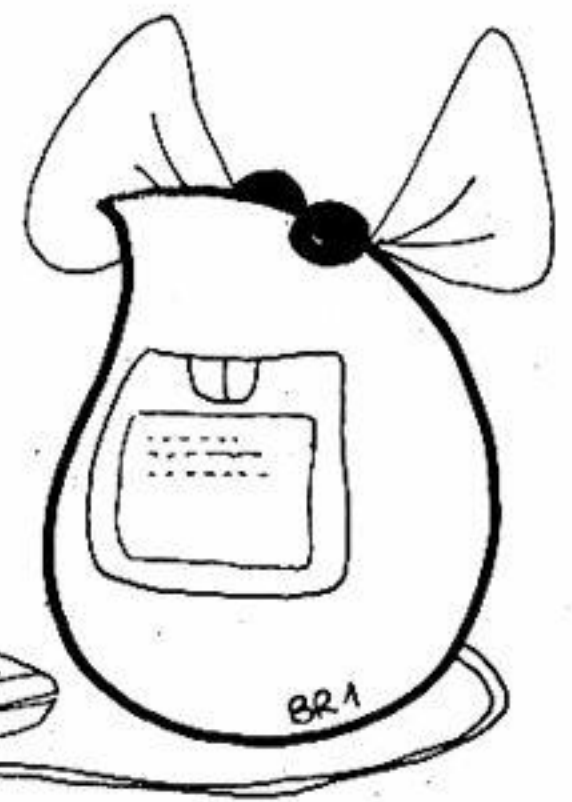
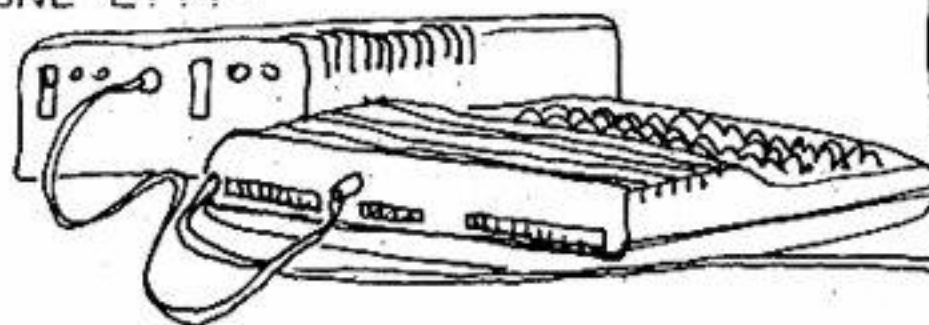



Immagine tratta dalla campagna pubblicitaria "Cagney & Lacey (Orion Television Syndacation)

DAL C/64 AL MONDO ESTERNO

La programmazione della porta utente consente il contatto con altri "universi"

di **Raffaele Borrelli**

Molti lettori hanno chiesto, più di una volta, come utilizzare la porta utente del Commodore 64. E' noto, infatti, che il dispositivo serve per collegare il computer con apparecchi che, al limite, possono essere comandati tramite la stessa porta.

In questo articolo, finalmente, verranno descritti i principi generali di funzionamento, grazie ai quali dovreste essere in grado di costruire dispositivi di comando per le più svariate applicazioni, come innaffiatori da giardino(!), elettrodomestici o luci psichedeliche, fino ad un massimo di 8 utilizzi contemporanei. Nei casi più semplici (e negli esempi di queste pagine) potremo accontentarci di collegare pulsanti o Led luminosi.

Sono, otto, infatti le linee che si possono utilizzare, e le possiamo individuare osservando la figura riportata sul manuale del C/64 (appendice I, pagina 143). Le linee sono contrassegnate dai simboli (presenti sempre sul manuale) da PB0 a PB7 e corrispondono ai piedini C, D, E, F, H, J, K, L della porta utente.

Come intuitivo, ad ogni piedino corrisponde un bit; gli otto bit, insieme, sono"comandati" da un byte, chiamato DDR (Data Direction Register: byte che controlla la direzione dei dati). Resettando (: ponendo a livello logico 0) alcuni dei bit citati, le corrispondenti linee PB0 - PB7 si configurano come ingressi. Ponendoli, al contrario a 1 (settandoli), le stesse linee verranno "viste" come uscite. Vi è quindi una locazione (la n. 56579), i cui bit governano le linee che costituiscono, a loro volta, la porta.

Supponendo che tutti i bit della locazione 56579 siano usati come ingressi, sarà poi possibile leggere lo "stato" della porta interrogando il computer con una banalissima...

Print Peek (56577)

Se, infatti, venisse applicata su tali pin un livello logico 0 (attenzione, però, a non superare MAI i 5 volt) i bit corrispondenti della locazione 56577 vengono posti a zero. Insomma, per programmare correttamente la porta utente abbiamo bisogno di DUE locazioni.

La prima (56579) serve a determinare quale linea deve essere considerata come ingresso (stato 0) oppure come uscita (stato 1).

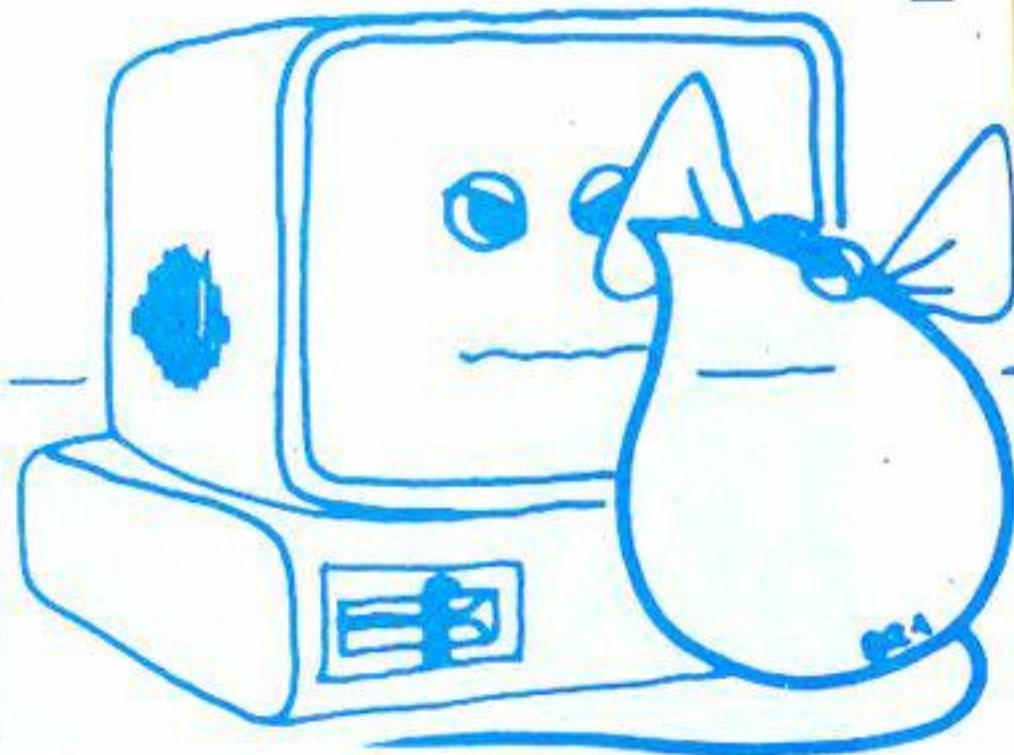
La seconda (56577), invece, rappresenta, bit per bit, i segnali effettivamente presenti sulle linee della porta utente (PB0 - PB7). In questo caso il computer, ovviamente, si limiterà a leggere la locazione 56577 e non si chiederà se un certo bit deve la sua "esistenza" grazie ad un dispositivo in ingresso o in uscita. Per sapere le origini di un bit che "vive" nella locazione 56577 dovremo interrogare la locazione 56579.

Nel caso di una linea configurata come ingresso, si potrà "riconoscere", ad esempio, la pressione di un pulsante. Il programma dimostrativo corrispondente attende che venga posto a 0 il bit della linea PB1 (collegata, appunto, ad un pulsante) per cambiare il colore del bordo.

Per i bit settati come uscita è possibile far comparire, sui piedini corrispondenti, un livello positivo di tensione (5 volt) oppure un livello zero (0 volt circa) sufficienti per illuminare (o spegnere) un led o eccitare un mini-relè, stando bene attenti a non esagerare con i carichi di corrente (non illudetevi di collegare direttamente il computer al trenino elettrico: brucereste tutto!).

Per illuminare il led del progettino riprodotto in queste pagine, bisogna quindi porre a 1 la linea PB0 (magari mediante un brutale Poke 54579,0) e settare a 1 il primo bit della locazione 56577; il programmino corrispondente sfrutta le locazioni citate per ottenere il lampeggio del led.

LE ALTRE LINEE

Volendo collegare un led su un'altra delle 8 linee disponibili (come la PB2) bisogna sempre ricordare di settare non più il bit corrispondente a PB0 (usato nei listati di esempio) ma quelli che servono.

Vediamo ora come ottenere, contemporaneamente, alcune linee in ingresso ed altre in uscita; per fornire un esempio faremo illuminare il led quando si preme il pulsante.

Bisogna quindi ordinare al computer di porre in ingresso la linea PB1 (sulla quale è montato il pulsante), ed in uscita la PB0 (dove è montato il led). Si interrogherà continuamente il secondo bit della locazione 56577, e si controllerà che se questo si azzeri (: pulsante premuto). Verificatosi tale evento, si dev porre a 1 il primo bit della stessa locazione, in modo da accendere il led.

Da notare che quando si setta una linea in modo "ingresso", il bit corrispondente della porta, comandato dalla locazione 56577, è abilitato solo per la lettura. Il terzo programmino esegue i controlli richiesti.

Si raccomanda vivamente, a chi non è pratico di montaggi di questo genere, di NON metter mano al saldatore in modo da non danneggiare il C/64: meglio rivolgersi a qualche amico esperto di montaggi elettronici.

Il montaggio, tuttavia, non è critico, grazie anche al modesto numero dei componenti impiegati.

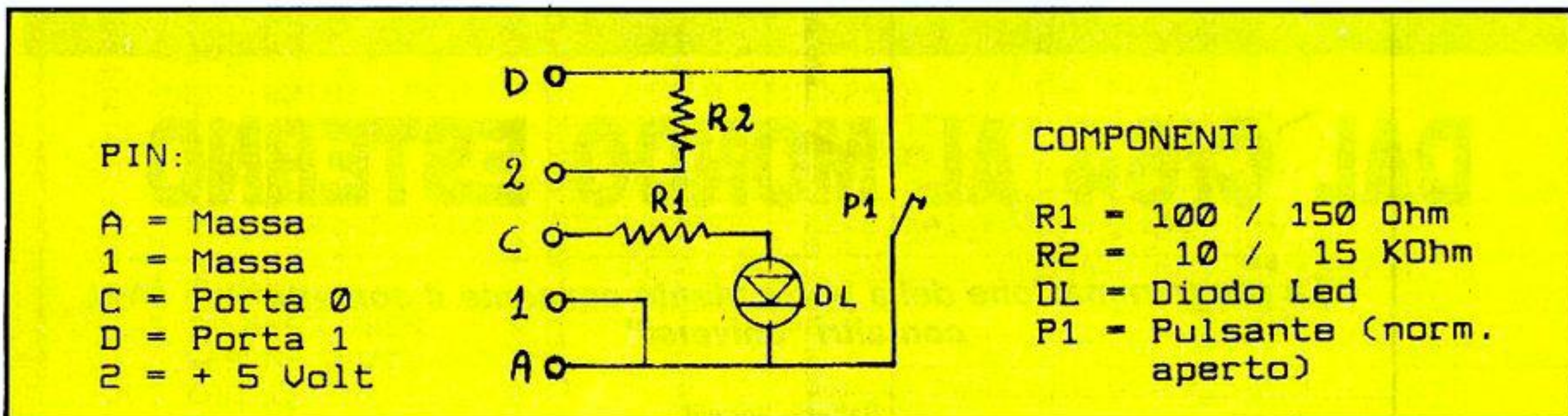
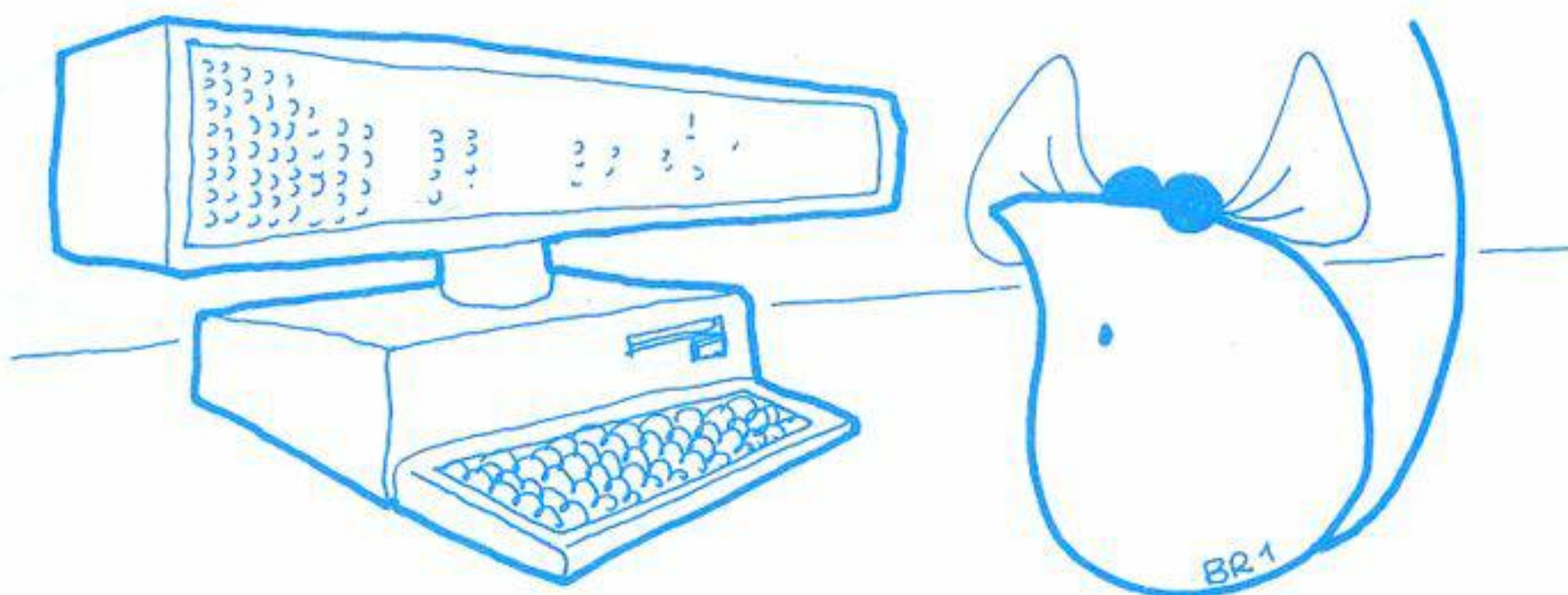
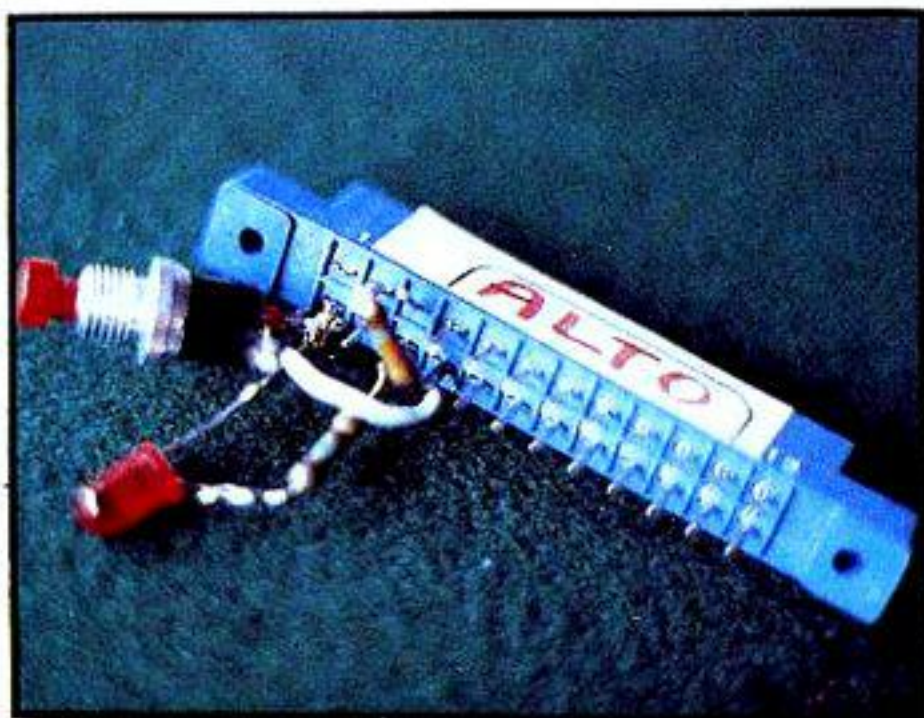
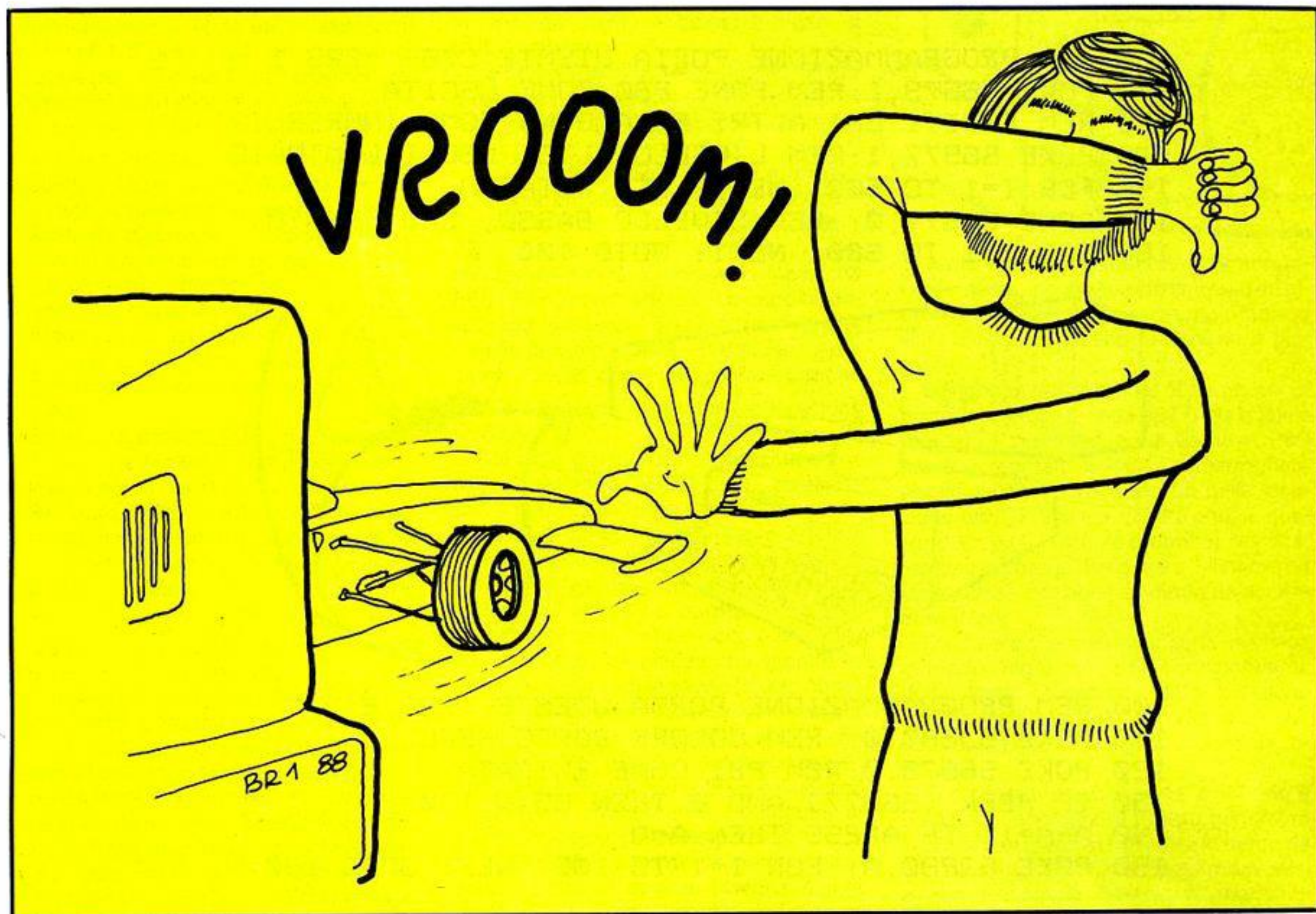
Da notare che possono verificarsi gravi malfunzionamenti nel caso in cui una linea, programmata come uscita (idonea, cioè, a pilotare un Led), venisse utilizzata come ingresso (collegamento ad un pulsante) o viceversa.

Rimane inteso che la Systems Editoriale declina ogni responsabilità nel caso dovessero verificarsi danni, anche se dovuti ad errori di stampa o di programmazione.

PIN:

A = Massa
1 = Massa
C = Porta 0
D = Porta 1
2 = + 5 Volt

COMPONENTI

R1 = 100 / 150 Ohm
R2 = 10 / 15 KOhm
DL = Diodo Led
P1 = Pulsante (norm. aperto)

```
100 REM PROGRAMMAZIONE PORTA UTENTE C/64. PRG 1
110 POKE 56579,1:REM PONE PB0 COME USCITA
120 REM (TUTTI GLI ALTRI RIMANGONO COME INGRESSI)
130 POKE 56577,1:REM LIVELLO ALTO, LED ILLUMINATO
140 FOR T=1 TO 500: NEXT: REM RITARDO
150 POKE 56577,0: REM LIVELLO BASSO, LED SPENTO.
160 FOR T=1 TO 500: NEXT: GOTO 130
```

```
100 REM PROGRAMMAZIONE PORTA UTENTE. PRG 2
110 POKE 53280,0: REM COLORE BORDO NERO
120 POKE 56579,0:REM PB1 COME ENTRATA
130 IF PEEK (56577) AND 2 THEN GOTO 130
140 A=A+1: IF A>255 THEN A=0
150 POKE 53280,A: FOR I=1 TO 100: NEXT:GOTO 130
```

```
100 REM PROGRAMMAZIONE PORTA UTENTE. PRG 3
110 POKE 53280,0 :REM SCHERMO NERO
120 POKE 56579,1:REM PB0 COME USCITA, PB1 COME ENTRATA
130 POKE 56577,0: REM LIVELLO BASSO, LED SPENTO.
140 IF PEEK(56577) AND 2 THEN 140
150 POKE 56577,1: POKE 53280,1: STOP
160 REM SE IL PULSANTE VIENE PREMUTO...
170 REM...ACCENDI IL LED E CAMBIA IL COL. DEL BORDO
```

CARATTERI ANIMATI

Manipolando l'interrupt è possibile realizzare semplici animazioni sul C/16 e sul Plus/4

di **Roberto Ferro**

Tempo fa, su C.C.C. è apparsa una simpatica routine per realizzare un particolare tipo di scrolling che si basava sull'interrupt e sulla ridefinizione dei caratteri. Ritorniamo questo mese ad occuparci dell'argomento, proponendo un nuovo tipo di manipolazione dei caratteri: l'animazione.

I FOTOGRAMMI

Chiunque si sia avvicinato, anche superficialmente, al mondo della cinematografia, saprà che l'animazione di un film è data dalla rapida successione di diversi fotogrammi ritraenti "quasi" la stessa scena, leggermente diversa dalla precedente.

La tecnica è tutt'oggi ancora tra le più usate per la produzioni di cartoni animati e animazioni computerizzate. I vari demo-animazione di Amiga, tanto per fare un esempio, sono nella maggior parte dei casi una serie di schermate presentate in rapida successione quanto basta per dare l'effetto di un movimento più o meno elastico. E' chiaro che un requisito fondamentale è la capacità di sostituire il più velocemente possibile le diverse immagini (e quindi ecco l'irrinunciabile ricorso all'assembly, escludendo il lentissimo Basic).

Sul C/16 e Plus/4 è possibile qualcosa di simile, ma solo al livello di un piccolo carattere. Utilizzando la tecnica di continua ridefinizione del carattere che abbiamo visto in "Shift Character" qualche tempo fa, possiamo rapidamente far apparire all'interno di un carattere, due o più immagini diverse che daranno l'effetto di un'animazione. Purtroppo esistono limiti ben evidenti in quanto è impossibile comporre una scena dettagliata all'interno di una misera griglia di 8 pixel per 8. Ciò non toglie, tuttavia, la possibilità di realizzare qualcosa di interessante come ad esempio un vorace Pac-Man oppure un omino che agita disperatamente le braccia, un mostriciattolo che muove le antenne e così via.

ANIMAZIONE CARATTERE

Il programma che proponiamo è, come sipuò ben vedere, molto breve. Prowede ad animare il carattere "A" con un profilo del personaggio più noto in assoluo nella storia del videogame: Pac-Man.

Anim Character è suddiviso in due routine. La prima ricopia la mappa dei caratteri ROM sulla RAM a partire dall'indirizzo $3800, mentre la seconda è la routine vera e propria, che "gira" in interrupt.

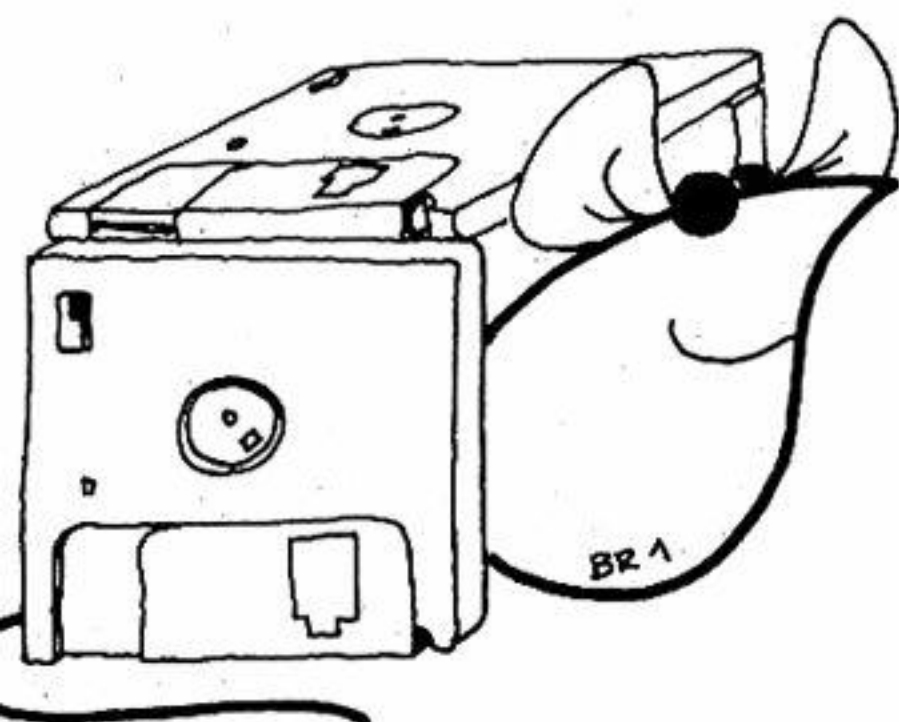
All'inizio vi è un ciclo di ritardo; poi, tramite una variabile posta in $D1 che assume alternativamente il valore 1 e 0, viene sostituito al carattere "A" una delle due posizioni di Pac-Man (bocca aperta - bocca chiusa) prelevate alternativamente a partire da $3F00 e $3F08, dove le aveva allocate il caricatore Basic.

La routine pubblicata è anche quasi la stessa che veniva utilizzata nel programma "My little Pet" apparso sul numero 16 di Software Club (in quel caso è stato, però, necessario animare ben più di un solo carattere!).

Il disassemblato, così come il programma Basic, è ampiamente commentato e non dovrebbero sorgere problemi di sorta nel comprenderne il funzionamento.

E' doverosa la solita precisazione: eventuali syntax error (o errori di altro tipo) possono mandare in tilt la macchina: evitateli accuratamente!

SUGGERIMENTI

Come abbiamo potuto vedere, la possibilità di ri-ridefinire i caratteri con l'IRQ consente di realizzare gli effetti più strani ed interessanti: merita quindi di essere compresa approfonditamente. Un suggerimento per creare routine sempre più utili può esere quello di animare più caratteri che, accostati, diano vita ad una figura più grande, come un uomo che cammina, un'auto con le ruote sterzanti o altro: basta avere un po' di pazienza, e molta fantasia, per disegnare nuovi caratteri.

```
10 REM **************************
20 REM      ANIM CHARACTER
30 REM   SOLO PER C/16 E PLUS/4
40 REM **************************
50 :
60 REM BY ROBERTO FERRO
70 :
80 SCNCLR
90 POKE 51,80:POKE 52,63
100 FORI=14208 TO 14309
110 READ A$:POKEI,DEC(A$)
120 CK=CK+DEC(A$)
130 NEXT I
140 IF CK<>11101 THEN 1280:ELSE CK=0
150 :
160 FOR I=16128 TO 16143
170 READ A$:POKEI,DEC(A$)
180 CK=CK+DEC(A$)
190 NEXT I
200 IF CK<>2346 THEN 1300
210 :
220 REM TRASFERISCE CARATTERI ROM IN RAM
230 SYS 14278
240 REM ATTIVA NUOVO SET DI CARATTERI
250 POKE65298,192:POKE65299,57
260 REM ATTIVA ROUTINE DI ANIMAZIONE
270 SYS 14265
280 PRINTCHR$(17)+" A  A  A  A  A  A  A"
1000 :
1010 REM ROUTINE PER ANIMARE IL
1020 REM CARATTERE 'A' CON PAC MAN
1030 :
1040 DATA E6,D0,A5,D0,C9,14,F0,03
1050 DATA 4C,0E,CE,A9,00,85,D0,A5
1060 DATA D1,C9,01,D0,12,A9,00,85
1070 DATA D1,A0,07,B9,00,3F,99,08
1080 DATA 38,88,10,F7,4C,0E,CE,A9
1090 DATA 01,85,D1,A0,07,B9,08,3F
1100 DATA 99,08,38,88,10,F7,4C,0E
1110 DATA CE,78,A9,80,8D,14,03,A9
1120 DATA 37,8D,15,03,58,60
1130 :
1140 REM ROUTINE PER RICOPIARE LA ROM
1150 REM DEI CARATTERI SULLA RAM ($3800)
1160 :
1170 DATA A2,00,BD,00,D0,9D,00,38
1180 DATA BD,00,D1,9D,00,39,BD,00
1190 DATA D2,9D,00,3A,BD,00,D3,9D
1200 DATA 00,3B,E8,D0,E5,60,00,00
1210 :
1220 REM DATI PER I DUE CARATTERI
1230 REM DI BOCCA APERTA E CHIUSA
1240 :
1250 DATA 3E,7F,F8,F0,B0,98,5F,3E
1260 DATA 38,7C,FE,FE,BE,9E,5C,38
1270 END
1280 PRINT"ERRORE NEL PRIMO BLOCCO DI DATA !"
1290 END
1300 PRINT"ERRORE NEL SECONDO BLOCCO DI DATA !"
```

UN'ENCICLOPEDIA PER IL TUO COMMODORE?


```
*------------------------------------*
*ROUTINE ANIM CHARACTER     per C/16 *
*by Roberto Ferro           & Plus/4 *
*------------------------------------*

3780  e6 d0     inc $d0   ;ciclo di
3782  a5 d0     lda $d0   ;ritardo
3784  c9 14     cmp #$14
3786  f0 03     beq $378b
3788  4c 0e ce  jmp $ce0e
378b  a9 00     lda #$00
378d  85 d0     sta $d0
378f  a5 d1     lda $d1   ;verifica il
3791  c9 01     cmp #$01  ;contenuto
3793  d0 12     bne $37a7;di $00d1 (0/1)
3795  a9 00     lda #$00  ;azzera $00d1
3797  85 d1     sta $d1
3799  a0 07     ldy #$07    ;ridefinisce
379b  b9 00 3f  lda $3f00,y;il carattere
379e  99 08 38  sta $3808,y;'A' come una
37a1  88        dey         ;bocca chiusa
37a2  10 f7     bpl $379b
37a4  4c 0e ce  jmp $ce0e;riprende IRQ
37a7  a9 01     lda #$01  ;pone
37a9  85 d1     sta $d1   ;a 1 $00d1
37ab  a0 07     ldy #$07    ;ridefinisce
37ad  b9 08 3f  lda $3f08,y;il carattere
37b0  99 08 38  sta $3808,y;'A' come una
37b3  88        dey         ;bocca aperta
37b4  10 f7     bpl $37ad
37b6  4c 0e ce  jmp $ce0e;riprende IRQ
37b9  78        sei
37ba  a9 80     lda #$80  ;devia
37bc  8d 14 03  sta $0314;interrupt
37bf  a9 37     lda #$37  ;per eseguire
37c1  8d 15 03  sta $0315;la routine
37c4  58        cli
37c5  60        rts


*------------------------------------*
*     ROUTINE TRANSFER CHARACTER     *
*------------------------------------*

37c6  a2 00     ldx #$00    ;preleva i
37c8  bd 00 d0  lda $d000,x;caratteri
37cb  9d 00 38  sta $3800,x;dalla
37ce  bd 00 d1  lda $d100,x;ROM ($d000)
37d1  9d 00 39  sta $3900,x;e li pone in
37d4  bd 00 d2  lda $d200,x;RAM ($3800)
37d7  9d 00 3a  sta $3a00,x
37da  bd 00 d3  lda $d300,x
37dd  9d 00 3b  sta $3b00,x
37e0  e8        inx
37e1  d0 e5     bne $37c8
37e3  60        rts
```

IL PIU' BRAVO DEL REAME

In periodi di (im)meritato riposo, può essere una buona idea verificare fino a che punto si conosce il proprio computer; per una sfida informatica all'ultimo sangue

Tutto ciò che si deve fare è rispondere correttamente ai quesiti, suddivisi in tre gruppi di difficoltà crescente, senza adoperare il computer.

Le soluzioni, sono pubblicate in altra parte di questo stesso fascicolo.

Nulla vieta che si tentino tutte le domande, ma, particolarmente se si è in compagnia, ecco un possibile regolamento.

Per ogni risposta esatta, indipendentemente dal livello, va assegnato un punto.

Per accedere al secondo livello (avanzato), bisogna totalizzare almeno 8 punti, mentre per passare dal secondo al terzo (top), occorre avere sommato almeno 13 punti, compresi quelli raggranellati nel primo livello.

In pratica, chi supera il primo ostacolo (10 quesiti) a punteggio pieno, potrà poi commettere 2 errori nel livello avanzato (5 quesiti) e passare ugualmente al Top Level.

Passando invece al secondo livello con il minimo (8), sarà poi costretto a rispondere a tutte e 5 le domande del livello avanzato, per accedere al Top.

Chi, dopo il terzo livello (altri 5 quesiti), avrà raggiunto o superato la soglia dei 17 punti, può tentare il Super Bonus, l'ultima e più difficile delle domande.

Vince, com'è ovvio, chi risponde esattamente al Super Bonus, o, se nessuno ce la dovesse fare, chi disporrà del punteggio migliore.

Se più di un concorrente dovesse risolvere il Super Bonus... controllate che qualcuno non abbia inventato un C/64 tascabile!

Vincitori e vinti, nonchè i "solitari", potranno poi dare un'occhiata ai commenti sul punteggio raggiunto, pubblicati a parte.

Ultima regola, alquanto ufficiosa: si squalifichi seduta stante chiunque tiri fuori il solito - Allegriaaaaaaa! -

PERCHE' UN QUIZ

Ebbene sì, è finalmente giunto il momento di scrollarsi di dosso la ruggine accumulata nei lunghi mesi trascorsi dall'ultima estate. Una ruggine che pervade, subdola, perfino il nostro Mega Processore Centrale (quello racchiuso da un cabinet volgarmente chiamato scatola cranica), troppo a lungo pressato da testi scolastici ed esami vari, o affettuosi consigli (di fantozziana memoria) da parte di direttori, capiufficio, e chi più ne ha più ne metta.

Insomma, senza ricorrere ad altri eufemismi, è tempo di vacanze.

Ed è anche il momento, triste per i più smanettoni, di riporre joystick e tastiera, resistendo alla insana tentazione di portare il C/64 in spiaggia o in montagna, magari per farlo abbronzare un po'.

L'ozio, d'altra parte, checchè ne dica il proverbio, non è poi tanto sgradevole.

E così, lacerato (si fa per dire) dalla contraddizione esistenziale tra il piacere della vacanza e la nostalgia di Print e Poke, qualcuno si ritroverà con la rivista in mano troppo lontano dalla più vicina presa di corrente.

Nel frattempo, però, perchè non dare una controllatina a quanto si è finora appreso, senza dover necessariamente ricorrere alla "macchina"?
Potrebbe essere un modo per trascorrere il tempo, e, se si è in compagnia di altri computerofili, ancora meglio: una sana competizione non guasta mai.

Anche i perdenti, in fondo, non avranno di che lamentarsi; sarà servito a colmare qualche lacuna.

Ed allora, con intonazione alla Frassica, tutti pronti per il quizzer!

PRIMO LIVELLO

1* Si consideri il seguente listato:

```
10 A = 3: B = 4: C = 5
20 D = A + B * C
30 E = (A + B) * C
```

Quale valore conterranno le variabili D ed E?

2* Supponiamo di non conoscere il valore assegnato ad una variabile, per esempio X. Provate a (de)scrivere una riga Basic che accerti se si tratta di un numero pari o dispari.

3* Prevedendo di dover utilizzare una schermata grafica in alta risoluzione 320x200, quante locazioni di memoria dovranno essere riservate per contenerla?

4* Come può essere simulata, assegnandone il valore ad una variabile, la rappresentazione di un numero casuale compreso tra 1 e 40?

5* Scrivete una procedura idonea ad individuare perchè la spia del disk drive ha cominciato (ipoteticamente) a lampeggiare.

6* Perchè un programma tipo...

```
10 FOR X = 0 TO 10
20 A(X) = X
30 PRINT A(X)
40 NEXT X
```

...funziona corrattamente, mentre sostituendo la riga 10 con...

```
10 FOR X = 0 TO 20
```

...si avrà un Bad Supscript Error In 20?

7* Come si ottiene, da programma, la visualizzazione del carattere "virgolette"?

8* Supponiamo che alla variabile A$ venga assegnata la stringa "commodore computer club". Dite (o scrivete), nella sua corretta sintassi, con quale comando è possibile estrarre, da A$, i caratteri del vocabolo "computer", assegnandoli (per esempio) alla variabile B$.

9* Qual è l'esatto significato (anche in italiano può andar bene) delle sigle ROM e RAM?

10* Dite esattamente cosa stamperà, dopo il Run, il seguente programma :

```
10 A = 4 : B = 8 : C = 10
20 X = A*C : X = X-A
30 Y = X - A * B
40 PRINT X / (Y - A)
```

LIVELLO AVANZATO

11* Dite, dopo un attento esame di queste righe Basic...

```
10 FOR X = 1 TO 100
20 PRINT "OKAY"
30 X = X-1 : NEXT
```

...quante volte verrà stampato "okay" sullo schermo.

12* E' possibile archiviare su disco dei dati SEQuenziali in un file di tipo PRG?

13* Un altro minilistato da esaminare:

```
10 FOR X = 1 TO 300
20 A$ = "*" + A$
30 NEXT X
```

Dopo un certo tempo dall'attivazione, il programma si arresterà con una segnalazione di errore: dite quale, ed il valore della variabile X a quel momento (deducibile con Print X).

14* Con quale comando è possibile visualizzare solo il nome del dischetto inserito nel drive, seguito dal numero dei suoi blocchi liberi?

15* Nella riga di programma che segue...

```
10 X = X + 1: PRINT X: GOSUB 10: RETURN
```

...una subroutine richiama se stessa, stampando ogni volta il valore di X. Per considerare valido il test, dite quale tipo di errore verrà segnalato, o, in alternativa, fino a quale valore giungerà X.

TOP LEVEL

16* Che cos'è, anche solo genericamente, un Side Sector?

17* Considerando l'organizzazione in memoria RAM di un programma Basic, a che cosa servono i primi 4 byte di ogni sua riga?

18* Ogni comando o funzione del Basic, viene trasformato dal sistema in un corrispondente codice numerico che lo identifica. Con quale termine inglese vengono comunemente definiti tali codici?

19* Qual è l'indirizzo di base dei registri predisposti alla gestione della grafica (VIC)?

20* Come si può ricavare, in Basic, l'esatta posizione in memoria di una variabile?

SUPER BONUS

* Dire quale risultato sortirà, in decimale, l'operazione booleana...

```
PRINT 255 OR 127
```

(Le risposte ai quiz a pag. 84)

CAMPUS

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

ESORDIENTI

- Tre problemi, tre soluzioni
- Linguaggio macchina, i vostri dubbi

ESPERTI

- 128/64 un biglietto A/R
- Immagini: piccole, piccolissime, grandi, grandissime

CAMPIONI

- La crescita dei caratteri

7 ft.

6 ft.

5

E' IN EDICOLA VR VIDEOREGISTRARE

VR
VIDEOREGISTRARE
MENSILE - ANNO V - N.42 - L. 5000

IN REGALO IL DIZIONARIO DEI TERMINI TECNICI

I MIEI PRIMI 50 ANNI: STORIA DELLA TELEVISIONE

OH VALENTINA: CREPAX DAI FUMETTI ALLA TV

DI TUTTO, DI PIU': UN VCR DA' FAVOLA

VIDEOTEST: QUATTRO APPARECCHI IN PROVA

Ssystems

LA PRIMA RIVISTA DI VIDEOREGISTRAZIONE ATTIVA

TRE PROBLEMI, TRE SOLUZIONI

La soluzione di tre problemi, posti da altrettanti lettori, può interessare anche voi

a cura di D. Pavone

STAMPA MINGA

☐ In ambiente Amiga Basic, non riesco a personalizzare l'output su stampante (MPS 1250). Se, per esempio, voglio stampare del testo in modalità espansa, secondo il manuale della stampante dovrei eseguire il comando Chr$(27) "w1", ma, inviandolo tramite il comando Lprint, ottengo solo la stampa dei caratteri "w1" senza altra modifica.

Gradirei sapere, inoltre, come collegare l'Amiga ad un televisore dotato di ingresso SCART, dal momento che ora utilizzo il modulatore.

Ho anche acquistato un cavo specifico, ma non ottengo alcun risultato.

(Roberto Lasio - San Sperate)

• Il problema, comune a tutte le stampanti collegabili ad Amiga, riguarda la modalità di invio dei codici di Escape (codice Ascii 27), che presiedono alla maggior parte delle funzioni di controllo sulle caratteristiche di stampa (vedi anche C.C.C. N. 62).

Mentre, infatti, risulta efficace inviare codici "normali" tramite Lprint (la stampa espansa, per esempio, è tranquillamente implementabile con Lprint Chr$(14)), per i codici di Escape è necessario aprire un canale di output con il comando Open, indirizzandolo verso la porta parallela.

In altre parole, tornando al caso della stampa espansa, occorrerà una miniroutine di questo tipo:

```
Open "par:" for output as 1
Print #1, chr$(27); "w"; chr$(1);
Print #1, "TESTO DA INVIARE..."
Close #1
```

...salvo, ovviamente, diversa sintassi legata al particolare tipo di stampante.

Sostituendo il Chr$(1) della seconda riga con Chr$(0), di solito si riabilita la stampa normale.

Per quanto riguarda il collegamento al TV tramite connettore Scart, la cosa è possibile solo se l'apparecchio video è in grado di commutarsi in monitor, e approntando i collegamenti necessari per l'audio, spesso non inclusi nei cavetti SCART in vendita.

Domanda d'obbligo: disponendo di un sistema Amiga degnamente ottimizzato, non è quantomeno poco coerente ricorrere ad un televisore?

QUESTIONE DI TAVOLOZZE

☐ In che modo si possono attivare i vari modi di risoluzione grafica?

(Fabrizio Conti - Roma)

• In ambiente Workbench, è sufficiente ricorrere al programma Preferences, la cui schermata principale mostra, in alto a destra, una opzione "Interlace", ed in basso a sinistra la scelta "Text".

Con Text, è sufficiente clickare su 60 oppure 80 per avere, anche uscendo dal programma con "Use", immediatamente disponibili le due risoluzioni orizzontali di 320 oppure, rispettivamente, 640 pixel.

Per disporre del modo Interlace, che raddoppia la dimensione verticale dello schermo (scelta On), è invece necessario salvare la configurazione su disco (opzione Save), quindi resettare il computer.

Da Basic le cose risultano addirittura più immediate: basta settare opportunamente l'ultimo parametro del comando Screen, secondo le modalità estesamente illustrate sul manuale.

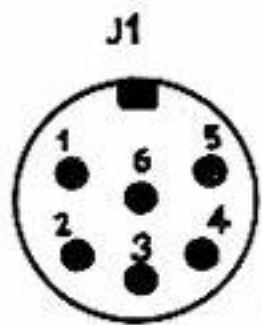
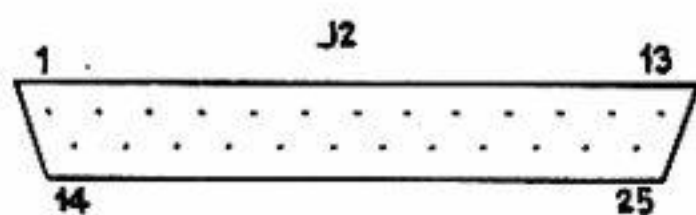
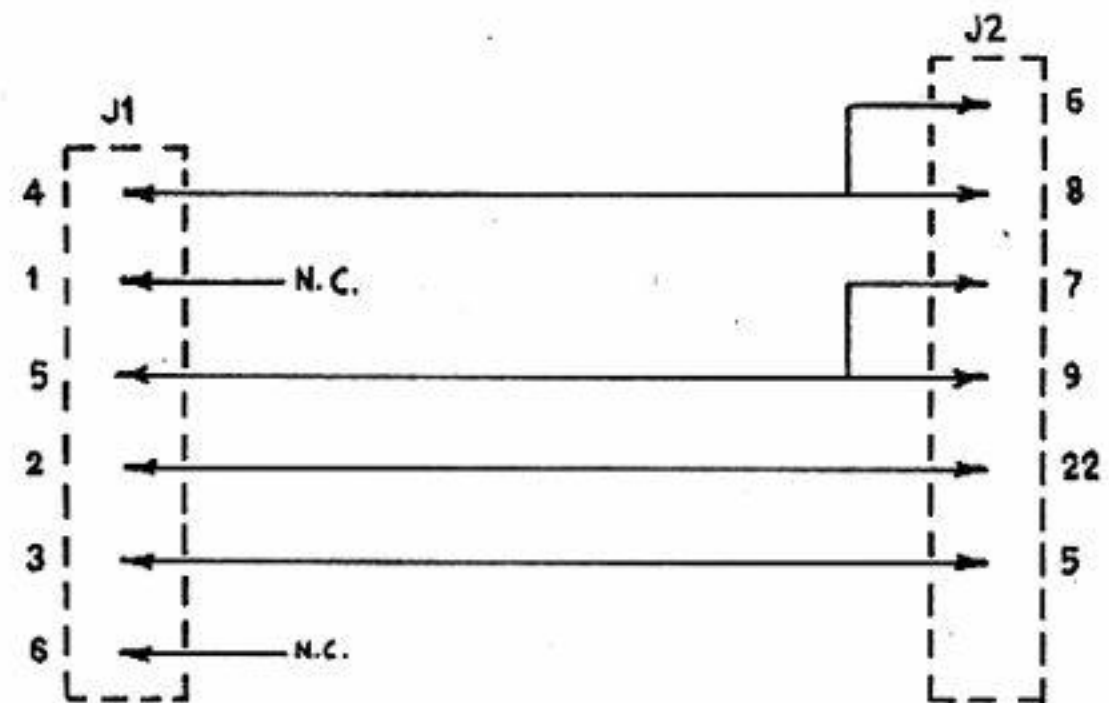
LE AVVENTURE DI
PRIMO GIOVEDINI
by Marco Mietta
ha collaborato Barbara De Toffoli
Un serio "inchiodamento". (2° File)

LATO DRIVE 1541/71/81

PORTA PARALLELA DI AMIGA

J1

VISTA FRONTALE

J2

VISTA FRONTALE

J1 = CONNETTORE DIN 6 POLI MASCHIO
J2 = CONNETTORE A VASCHETTA 25 POLI MASCHIO

A titolo di esempio, si digitino queste poche righe...

Screen 1, 640, 400, 1, 4
Screen 2, 320, 400, 1, 3
Screen 3, 640, 200, 1, 2
Screen 4, 320, 200, 1, 1

...e si lanci il programma.

Portandosi ora con il cursore sulla barra del primo schermo (vuoto) che appare, lo si tiri giù premendo il pulsante sinistro del mouse e contemporaneamente muovendolo verso il basso.

Apparirà il secondo schermo, ma di risoluzione diversa dal primo, come si può constatare dalle dimensioni del gadget di priorità.

Si ripeta la stessa operazione altre due volte, e si ammireranno i due schermi in Interlace.

Per tornare alla normalità, si attivi (agendo sui gadget di priorità, o tirando giù anche l'ultimo schermo interlace) la finestra di output del Basic, e si chiudano gli schermi digitando in modo diretto...

Screen close 1
Screen close 2
Screen close 3
Screen close 4

EMULANDO SI IMPARA

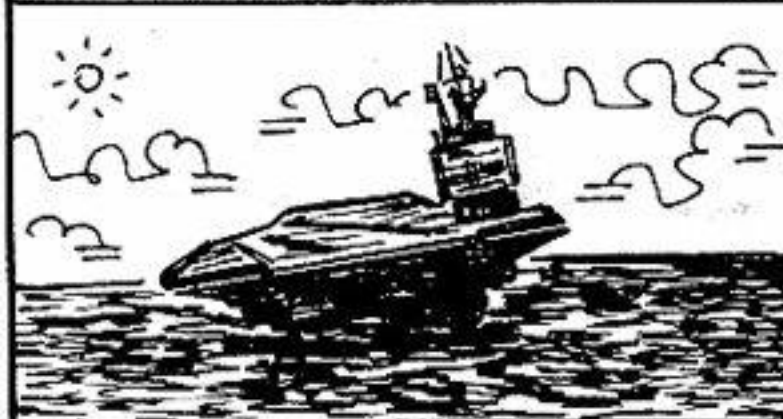
□ Ritengo che lo schema di collegamento per l'Emulator 64 (descritto da Roberto Ferro su C.C.C. N. 60, pagina 26), contenga alcune imprecisioni.

Penso che il motivo sia da ricercare nella errata indicazione della zoccolatura nel connettore DIN a 6 poli, definita come "vista frontale", mentre dovrebbe essere "lato saldature".

Corretta, invece, è la parte destra della figura, che illustra lo schema dei collegamenti.

Preciso che la porta parallela dell'Amiga sembra funzionare egregiamente, se collega-

RIASSUNTO DEL FILE PRECEDENTE :
A causa di un incredibile errore nel settaggio di alcuni puntatori, la portaerei di Primo Giovedini si è diretta su di un banco di sabbia (locato a $4000/$7FFF), incagliandosi...

... e rimanendo seriamente sbandata sulla sinistra.

Primo Giovedini e Jack Assembler sono così stati incaricati dal comandante C.P.U. di coordinare il lavoro della squadra di salvataggio: fra innumerevoli difficoltà e molti break pubblicitari si riesce ad elaborare un piano...

ta ad una stampante, ma vorrei sapere se tale errore può aver causato danni.
(Francesco Mollica - Taranto)

• Ebbene, ha ragione il nostro Francesco.

Purtroppo, per una imperdonabile svista dell'autore dell'articolo citato, la figura in questione avrebbe dovuto mostrare il connettore DIN come appare in queste pagine.

L'inversione dei piedini 1-5 e 2-4, a parte il mancato funzionamento, non dovrebbe comunque poter causare danni maggiori, in quanto non coinvolge tensioni elettriche di alimentazione.

Questo per rispondere anche ad un altro lettore (dalla firma quasi illegibile: Roberta Verucchino?) che lamenta invece, a seguito della realizzazione del cavo, danni non solo al computer, ma addirittura all'audio del monitor.

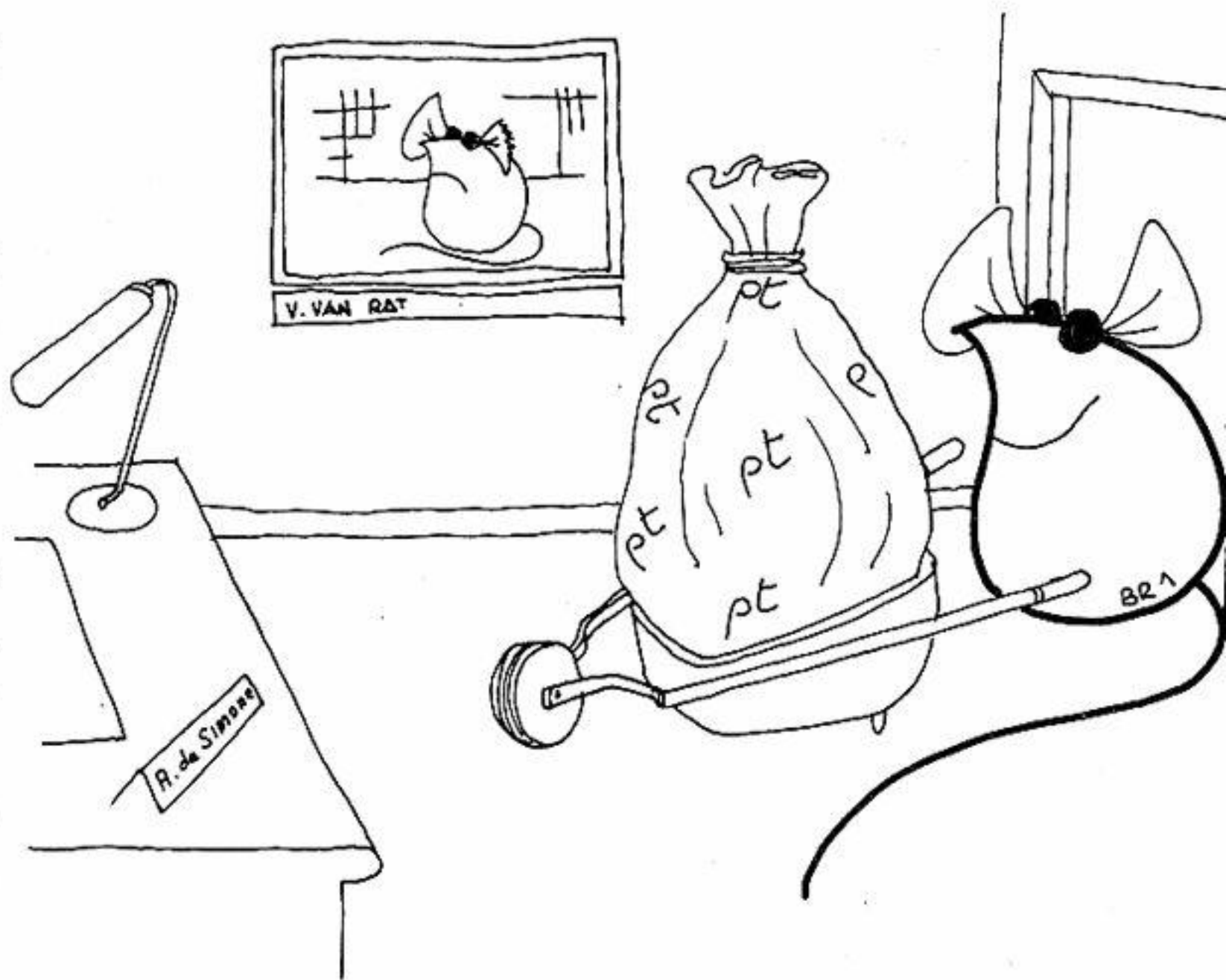
In questo caso, verosimilmente, i guasti sono da imputarsi ad errati (o indesiderati) contatti nel connettore parallelo a 25 poli, il tutto peggiorato (come descritto nella lettera) da un'accensione del drive 1571 DOPO aver attivato il sistema Amiga - "C/64 emulator".

Per i curiosi aggiungeremo che, anche con i cavetti originali, la prima versione dell'emulatore (cui fa riferimento l'articolo) dava dei problemi, in parte risolti dalla comparsa di "Emulator 2".

Nella versione più recente lo schema dei contatti hardware è stato leggermente modificato, come rilevabile dalla figura (stavolta) corretta.

In particolare, è stato sconnesso il piedino 6 (Reset) di J1, mentre il piedino 2 (sempre di J1) è collegato alla "terra" della porta parallela tramite il suo piedino 22 (anche il 20, tuttavia, è un segnale di terra).

Non resta, da parte nostra, che procedere al rituale cospargimento di cenere sul nostro capo che, se confrontato con quanto avverrà nei confronti di ISSO, le purghe staliniane sembreranno uno scherzo.

Un'ultima nota è doverosa nei confronti di coloro che, avendo realizzato l'interfaccia, non l'hanno vista funzionare: per forza! Funziona solo con il programma "Emulator 64" e non da sola.

POSTAMIGA

Le domande relative al mondo "Amiga" devono essere indirizzate a:

Commodore Computer Club
Postamiga
Via Mosè 18
20090 OPERA (Mi)

Si consiglia di scrivere in modo chiaro (meglio se a macchina) e di non porre troppi quesiti contemporaneamente.

Non ci è possibile rispondere privatamente, nemmeno se vengono acclusi francobolli per la risposta.

LINGUAGGIO MACCHINA, I VOSTRI DUBBI

Ecco le risposte ad alcuni vostri quesiti

di **Domenico Pavone**

MACRO DIFFICOLTA'

□ A volte, usando il Macro Assembler della Commodore (per C/64), al termine dell'assemblaggio, anche se non si sono commessi errori, compaiono strani massaggi...
(Claudio Fontacci - Roma)

• "Ad esempio, (continua il lettore)...

```
ERRORS = 0
END OF ASSEMBLY
:0000180018
```

Quando poi cerco di caricare in memoria il file oggetto, il Loader si blocca, la spia del drive rimane accesa, e per tornare al cursore devo premere Run/Stop.

Naturalmente, anche impartendo la Sys al mio programma, questo si blocca."

Il problema, se ci fai caso, riguarda soprattutto i programmi che contengono, in coda alla routine digitata con l'editor, alcuni codici non corrispondenti ad istruzioni Assembly (come, ad esempio, un messaggio scritto in codici Ascii).

L'inconveniente lamentato è dovuto alla mancanza dell'istruzione .END (col punto iniziale!), che DEVE concludere la stesura del file sorgente.

Si badi che tale istruzione deve proprio essere l'ultima, indipendentemente dalla fine "logica" del programma Assembly.

In effetti, talora può essere omessa e tutto funziona ugualmente, ma è buona norma, specialmente se non si è già superesperti, aggiungere sempre .End in coda al disassemblato.

IL TEMPO E IL CASO

□ Vorrei sapere come si legge il valore del timer del C/64 in linguaggio macchina, e se esiste (e come si usa) una routine in Rom che genera numeri casuali.
(Maurizio - Roma)

• L'argomento della prima domanda è stato abbondantemente trattato nel n. 61 (pagina 86) della rivista, per cui passiamo direttamente alla seconda.

Dal momento che il Basic contempla la funzione Randomize (RND), ovviamente deve esistere una routine dell'interprete che la gestisce.

Questa, per l'esattezza, si trova memorizzata a partire dalla locazione 57495, in esadecimale $E097.

Quando viene richiamata (da una Sys, o, in Assembly, da Jsr), attiva una varie procedure che, troppo complesse per essere chiarite in questa sede, in ultima analisi memorizzano un valore reale compreso tra 0 ed 1 nel cosiddetto Fac1 (= Accumulatore in virgola mobile), identificato dalle locazioni da 97 a 104.

La cosa è facimente verificabile impartendo direttamente...

sys 57495: sys 49599

La prima routine di sistema crea il valore casuale, mentre la seconda si limita a stampare, sullo schermo, il contenuto di Fac1.

Ripetendo più volte l'operazione si vedrà come i numeri saranno sempre diversi.

Per un uso più sofisticato, data la particolare notazione (in virgola mobile) dei valori trattati, occorre utilizzare le routine aritmetiche contenute nella Rom del 64 (vedi n. 53, pagina 61), basate sulla manipolazione dei due Fac disponibili.

Come esempio, ecco due brevissime routine che

generano, e stampano, un valore casuale intero compreso tra 0 e 100. In Basic...

```
Print Int (Rnd(0) * 100)
```

..e in Assembly...

```
JSR $E097 Randomize.
JSR $BC0C Fac1 in Fac2.
LDA #$64  Numero 100
JSR $BC3C in Fac1.
JSR $BA2B Fac1 * Fac2.
JSR $BCCC Intero di Fac1.
JSR $BDD7 Stampa FAC1.
RTS       Return.
```

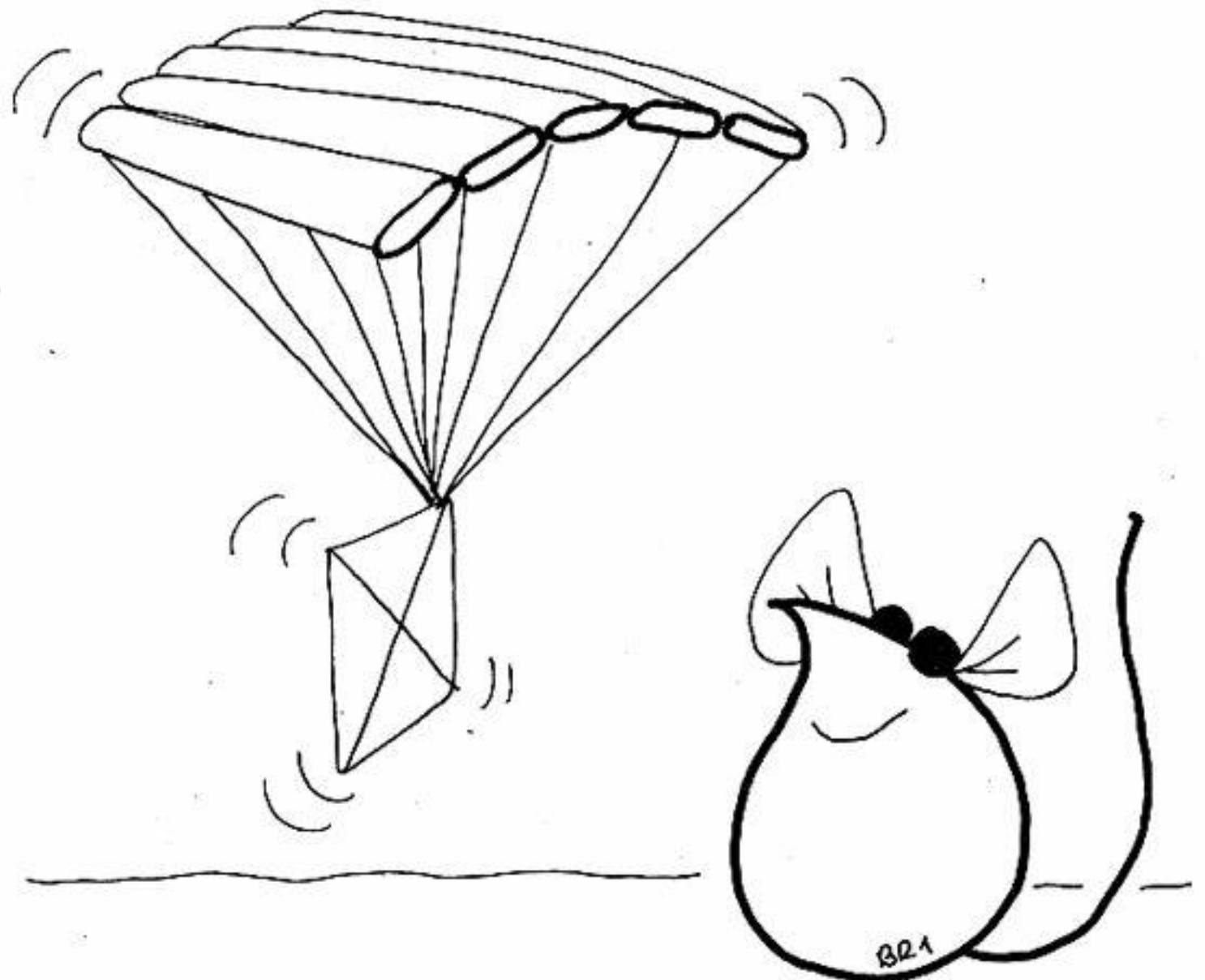
C/128, TROPPO FORTE

☐ **Sono un ragazzo di 14 anni, passato dal Plus/4 al C128. Potreste realizzare un articolo con la descrizione delle varie locazioni di Input/Output ($D000 - $E000)? E ancora: quali sono le locazioni che individuano i due microprocessori CIA 1 e CIA 2?**
(Federico Alpi - Roma)

• Gli argomenti che proponi sarebbero sufficienti ad occupare un'intera annata della rivista, sempre che questa si occupasse solo del C/128: altro che un articolo!

Per rendere meglio l'idea, ecco cosa è contenuto nell'area riservata all'I/O, con le locazioni di partenza di ogni gruppo di registri (la risposta alla seconda domanda è implicita):

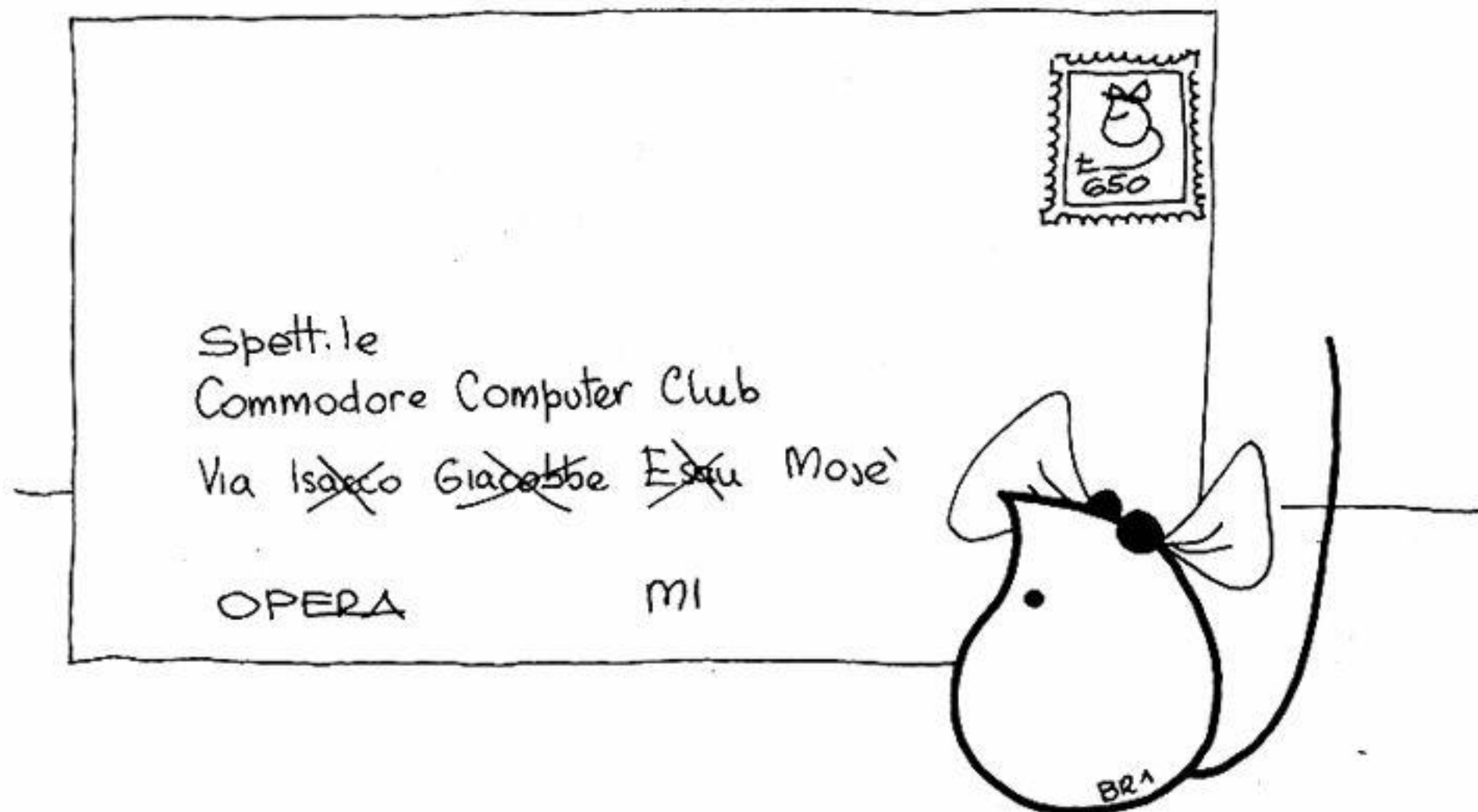
53248 ($D000) = Vic (gestione 40 colonne).
54272 ($D400) = Sid (gestione del suono).
54528 ($D500) = Mmu (configurazione della memoria).
54784 ($D600) = Vdc (80 colonne).
55296 ($D800) = Ram colore.
56320 ($DC00) = Registri del CIA 1.
56576 ($DD00) = Registri del CIA 2.
57088 ($DF00) = Reg.controllo eventuali espansioni.

Per soddisfare le tue sane curiosità non ti resta che continuare a spulciare tra le righe dei vari articoli già pubblicati, o che lo saranno, sempre che Amiga non decida di far fuori le minoranze... etniche (= i centoventottisti).

Nel tuo caso, comunque, non dovrebbero esserci grossi problemi: ai cambiamenti in meglio sei già abituato.

C/128 COME C/64

☐ **E' possibile convertire per il C/128 la routine di ordinamento veloce di una matrice, apparsa sul numero 59 della rivista? In caso affermativo, dove andrebbe allocata e a quali indirizzi corrispondono quelli dell'interprete del basic del C/64...(segue elenco)**
(Santi Innaimi - Palermo)

• La conversione di una routine in linguaggio macchina da C/64 a C/128, pur se fattibile, non è così automatica come sembra.

In altre parole, non basta sostituire qualche indirizzo di salto per adattarla ad un elaboratore strutturalmente molto diverso.

Nel caso della routine citata, per esempio, la manipolazione delle variabili basic deve necessariamente tenere presente che queste, nel C/128, si trovano in banco uno, per cui sono accessibili da programma solo attraverso apposite routine (di sistema) di lettura/scrittura tra banchi diversi.

Per quanto riguarda poi le routine dell'interprete, è chiaro che ne esistono di analoghe in quello del 128, ma questo non vuol dire che siano "identiche".

Anche qui, non è sufficiente modificare l'indirizzo di una JSR per avere lo stesso effetto nei due computer, in quanto sarebbe necessario cambiare anche il modo in cui i dati sono gestiti dopo il salto.

Occorre, insomma, anche sulla base di uno stesso algoritmo, conoscere molto in profondità le differenze tra i due computer per procedere ad una conversione... e sempre che ne valga la pena: non è certo infamante usare il C/64 incorporato nel 128.

Il problema dell'allocazione, invece, è di semplice soluzione.

Se lo spazio è sufficiente, si può adoperare tutta l'area che va da $1300 ad $1BFF; in caso contrario, basta impartire un comando Graphic1 seguito da un Graphic0, per avere libera ed usabile tutta la sezione di memoria da $1C00 a $3FFF di banco 0.

Attivando, infatti, la pagina grafica, il basic viene spostato a partire da $4000, e, purchè non venga utilizzata l'alta risoluzione, lo spazio antecedente (libero da interferenze) diventa ideale per memorizzarvi programmi in linguaggio macchina.

L'invito, comunque, è sempre lo stesso: trarre le opportune informazioni dagli articoli pubblicati, anche se apparentemente trattano argomenti diversi da quello che si intende sviluppare.

INTERRUZIONI 128

☐ Ho letto su un'altra rivista del settore che alla locazione 53295 è gestito l'interrupt della tastierina numerica del C/128, purtroppo senza altri chiarimenti. Come può essere usata detta tastierina in modo 64?
(Enrico Colantuono - Venticano)

• Rivolgendosi al nostro servizio arretrati.

Su "questa" rivista, infatti, sono già state pubblicate ben due routine che rispondono al quesito, sui numeri 35 (pagina 34) e 38 (pagina 69).

128/64, UN BIGLIETTO DI ANDATA E RITORNO

Una routine in l.m. per passare dal modo 128 al C/64, e viceversa, senza perdere il programma Basic in memoria

di **Domenico Pavone**

Il manuale del C/128, sin dalle prime battute, ci introduce alla principale delle sue caratteristiche: quella di essere uno... e trino.

Tralasciando irriverenti accostamenti divini, limitiamoci a constatare la presenza, nella stessa macchina, di ben tre computer: due della stessa famiglia (C/64 e C/128), ed uno decisamente alieno (il CP/M).

Continuando a sfogliare il libretto d'uso, si fa sempre più strada la convinzione che si tratta di tre mondi chiusi in se stessi, assolutamente indipendenti uno dall'altro e, soprattutto, senza la minima possibilità di rapporto (qualcuno, un tempo, avrebbe parlato di "monadi").

La cosa può non destare meraviglia per il cp/m, che sfrutta addirittura un diverso microprocessore (lo Z-80), ma per il modo 128 ed il C/64 qualcosa in più della stretta parentela deve pur esserci, come possiamo subito verificare.

Si provi a digitare, in modo 128...

Poke 8192, 55

...(e Return) in modo da scrivere il valore 55 in una locazione ben determinata (la n. 8192).

Ci si porti ora in modo 64 (go64), e si impartisca...

Print Peek (8192)

Sullo schermo apparirà proprio il valore 55, rimasto invariato nonostante il cambio di sistema operativo.

Torniamo ancora in modo 128 (tramite il tastino di reset) e, grazie ad un nuovo Print Peek (8192), si potrà constatare come il contenuto di questa locazione sia sempre 55.

Esistono, dunque, delle porzioni di memoria in comune tra i due computer, che non vengono alterate nel passaggio da una modalità all'altra.

E fin qui niente di trascendentale, almeno per chi possiede qualche cognizione di linguaggio macchina.

E' infatti noto che, allocando in dette aree programmi LM, questi possono essere utilizzati da entrambi i computer, sempre che non richiamino esclusive routines di sistema o necessitino (per il 128) di particolari configurazioni di banco.

Ma quali sono esattamente queste aree?

E, cosa più importante, possono essere sfruttate anche per trasferire automaticamente un programma Basic da una modalità all'altra?

Inutile aggiungere che una simile possibilità risulterebbe alquanto comoda.

Immaginate di utilizzare il super editor del 128 per stilare (o modificare) un programma del C64, per poi collaudarlo passando direttamente a quella modalità, senza tutti i noiosi salvataggi intermedi.

Se poi qualcosa non funzionasse a dovere, nessun problema.

Basterebbe tornare tranquillamente al 128 e, sempre con il programma in memoria, apportare le modifiche del caso.

Non sarebbe proprio male, ma... è fattibile?

Risposta: sì, ed in maniera neanche troppo difficile.

Il passaggio dal modo 128 al modo 64 è possibile grazie ad una routine l.m.

Dopo parecchi minuti, però...

Sys, non vedo i risultati: tutto quello che ha fatto...

... è stato ruotare i pixel della bandiera! Un po' poco!

Scusa, capo...

... ma lo sai che il Basic è lentino!

A! L'avete fatta in Basic ?!??

RAPPORTO TRA DIVERSI

Nell'ambito delle aree "comuni" descritte in queste pagine, ve ne sono alcune che, pur non interessando direttamente l'interscambio tra programmi Basic, possono tuttavia tornare molto utili.

Tutti e due i sistemi, come si può rilevare dalla solita figura 1, dispongono di una porzione di memoria libera da interferenze: il C/64 da $C000 a $D000, il C/128 da $1300 a $1C00.

Entrambe, pur se utilizzate diversamente nell'ambito del proprio sistema, sono pienamente condivise da C/64 e C/128.

Il fatto però che (per esempio) l'area da $C000 nel C/128 sia accessibile dal Basic, non vuol dire che non possa essere utilizzata per scambi di natura diversa (leggi: linguaggio macchina), tra l'altro senza ricorrere a particolari accorgimenti.

Un esempio banale: se, in modo 128, si carica con Bload un programma LM del C/64 che inizia (classicamente) da $C000, e poi ci si porta in modo 64, una sys 49152 attiverà normalmente il suddetto programma, dimostrandone l'effettiva presenza.

Lo stesso discorso, ovviamente, vale per routine direttamente assemblate con il monitor del C/128.

Come esempio, si entri in ambiente Monitor e si digiti A C000 (:assembla da 49152 in poi), seguito da questa miniroutine (il Return va dato dopo ogni istruzione):

```
LDA #$0
STA $D020
STA $D021
RTS
```

Penserà il Monitor, dopo ogni Return, ad aggiornare automaticamente l'indirizzo di ogni istruzione.

Ora si prema il tasto Commodore, e quindi il reset.

Una volta entrati in modalità C/64, si digiti Sys 49152, e lo schermo assumerà il colore nero per lo sfondo ed il bordo.

In pratica è come se disponessimo del monitor anche per il C/64: più comodo di così...

Le stesse condiderazioni, ovviamente, valgono anche per l'area da $1300 a $1C00, come del resto dimostrato dalla routine di queste pagine.

Il tasto Restore viene usato come "interruttore" per il trasferimento

UNIVERSI PARALLELI

Cominciamo col delimitare i termini del problema, dando un'occhiata alla organizzazione generale della memoria dei due sistemi, che potete vedere schematizzata in figura 1.

Per il C/128, come si può notare, viene preso in considerazione solo il banco 0, e questo per due motivi.

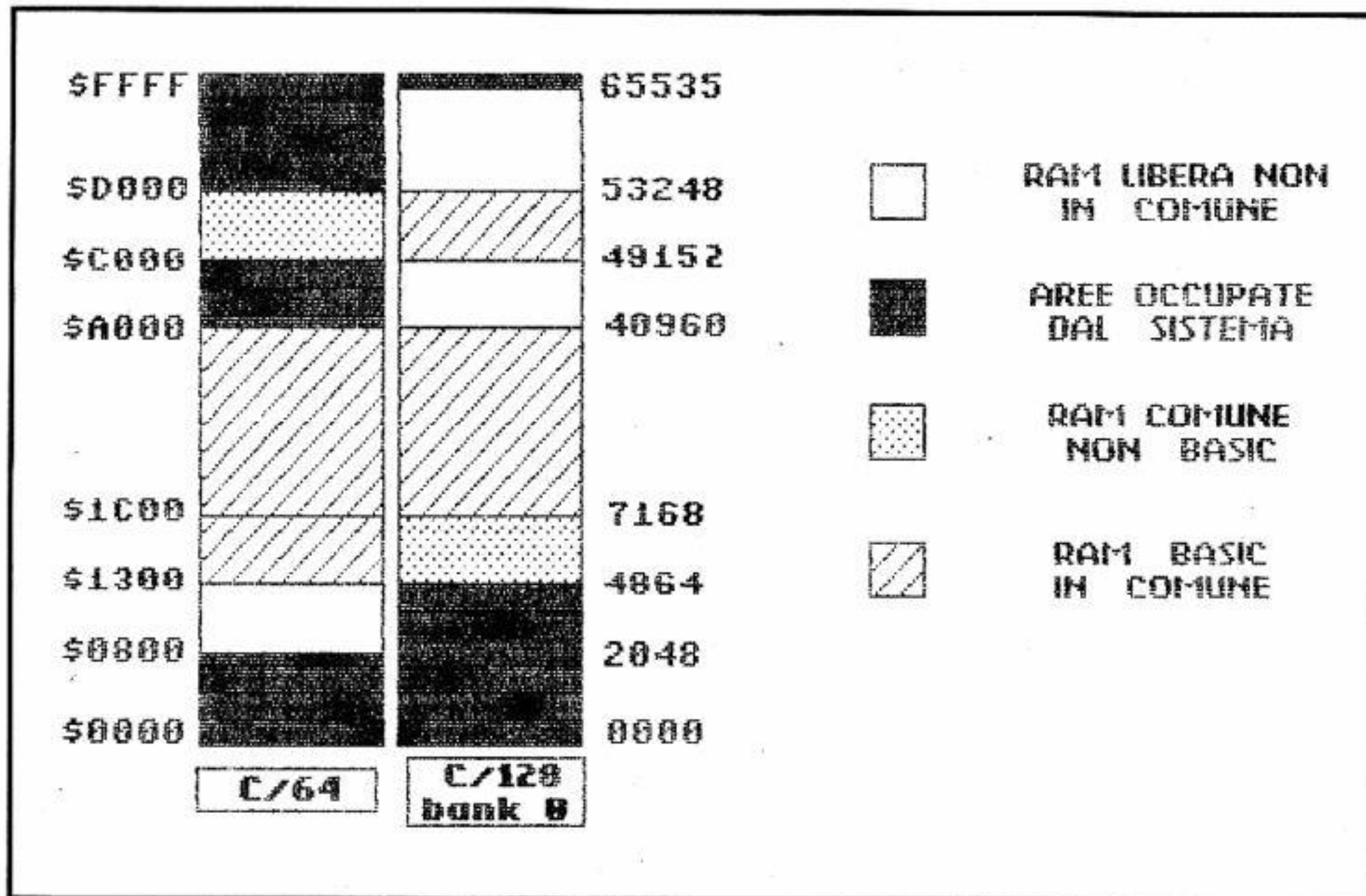
Anzitutto perchè è adibito a memorizzare il testo dei programmi Basic, ma anche perchè, quando si passa in modo 64, è il suo contenuto che viene condiviso, limitatamente alle aree comuni tra i due tipi di computer.

Questo, tra l'altro, senza la necessità di impartire alcun comando Bank 0.

Tornando alla figura 1, la prima ovvia differenza tra C/64 e C/128 è che, ad eccezione delle aree occupate dal sistema operativo, quasi tutta la memoria di bank 0 è disponibile per il Basic, mentre nel C/64 questo viene immagazzinato da $0800 ad $A000 (i corrispondenti decimali sono ricavabili dalla figura stessa).

Da tenere presente che, nel caso del C/64, le variabili seguono a ridosso del testo del programma.

$FFFF 65535
$D000 53248
$C000 49152
$A000 40960
$1C00 7168
$1300 4864
$0800 2048
$0000 0000
C/64
C/128 bank 0

RAM LIBERA NON IN COMUNE
AREE OCCUPATE DAL SISTEMA
RAM COMUNE NON BASIC
RAM BASIC IN COMUNE

mentre il C/128 dispone di un altro banco (1) per memorizzarle.

Sempre dalla stessa figura, si deduce che il "territorio" Basic condiviso al cento per cento è quello da $1C00 ad $A000.

Il che, tradotto in pratica, significa che se carichiamo un programma Basic dal C/128 (di dimensioni non eccedenti le locazioni in comune), dopo il passaggio al C/64 questo sarà integralmente presente nella RAM da $1C00 in poi, anche se non direttamente utilizzabile.

Viceversa, caricando un programma mentre ci si trova in modo 64, dopo il passaggio in modo 128 ne verranno perse alcune parti, dal momento che il nuovo s.o. reinizializza alcune locazioni comprese tra $800 e $1C00.

Come forse si sarà intuito, la soluzione del problema consiste nel trasferire opportunamente le diverse aree di memoria prima o dopo il cambiamento di "modo"; ma non è proprio come bere un bicchier d'acqua.

Inutile aggiungere che, per il tipo di lavoro da svolgere, si dovrà operare in linguaggio macchina.

E poichè non tutti i lettori gradiscono gli approfondimenti didattici, prima di addentrarci nell'argomento, vediamo all'opera una routine che ci consenta quanto sinora considerato impossibile: uno "switch" nei due sensi tra C/128 e C/64, con programma Basic... al seguito.

UN POSSIBILE CONNUBIO

Si resetti, dunque, il computer e, badando che sia configurato in modo 128, si copi il listato di queste pagine.

Dopo averlo salvato su disco (utilizzando la portentosa periferica denominata "Drive", frutto delle più recenti conquiste tecnologiche) o nastro (mefitico avanzo della passata civiltà paleopastorale), lo si attivi con Run.

Dal momento in cui riapparirà il Ready, la nuova implementazione risulterà disponibile.

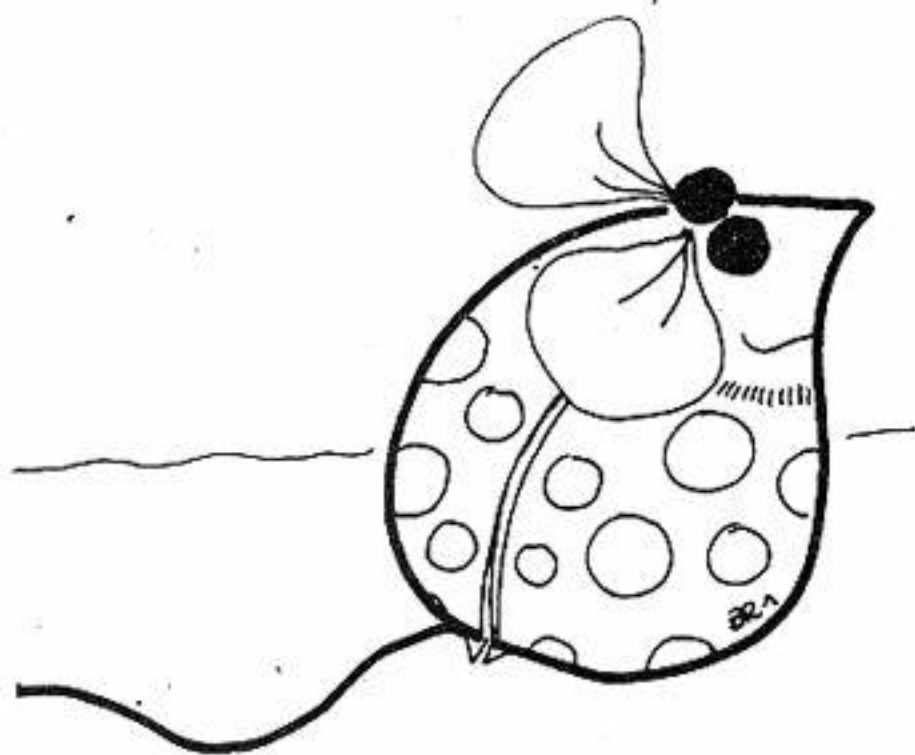
Vediamo subito come va usata.

Si impartisca New (e Return), quindi si digiti la riga di programma...

Un programma l.m. presenta meno problemi per esser trasferito da 128 a 64

... ma noi siamo scienza, non fantascienza!

... e comunque, per gli effetti speciali...

... l'esperto di RASTER costava troppo !!!

PUNTATORI DEL BASIC UTILIZZATI

	C/64	C/128
Inizio del testo	\$2B-\$2C (43-44)	\$2D-\$2E (45-46)
Fine del testo	\$2D-\$2E (45-46)	\$1210-\$1211 (4624-4625)
Tetto del basic	\$37-\$38 (55-56)	\$1212-\$1213 (4626-4627)
Base stringhe	\$33-\$34 (51-52)	(*)

(*) Non significativi in questa sede, in quanto riferiti al banco 1.

Se non disponete di un monitor l.m. per C/64 potete usare tranquillamente quello del C/128

10 Print "Partenza dal C/128"

Ci si porti ora in modo 64 con Go64 (o premendo il reset unitamente al tasto Commodore), e si impartisca List: ecco lì la nostra brava riga di programma, assolutamente reale (si provi con Run!), come se nulla fosse accaduto.

Collaudiamo ora il passaggio inverso, non così automatico come quello appena visto.

Prima, però, per constatare il corretto funzionamento del sistema, aggiungiamo (sempre in modo 64) un'altra riga al nostro miniprogramma:

20 Print "passato dal C/64"

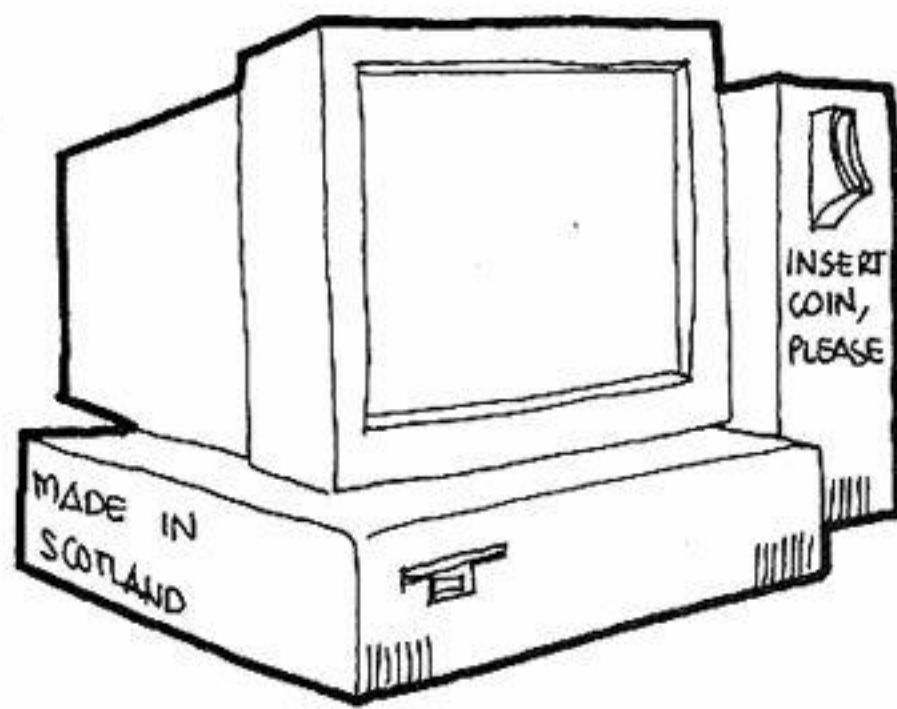
Per tornare al 128, occorre seguire, nell'ordine, questa semplice procedura:
anzitutto, premere il tasto Restore (una sola volta!). Un messaggio segnalerà la successiva operazione, che consisterà nella consueta pressione del tastino di reset.

Si badi, in questa fase, che il drive non contenga un disco predisposto (tramite autoboot) al caricamento in memoria di qualche altro programma in area Basic: il listato proveniente dal C/64 andrebbe ovviamente perduto.

Dopo che il computer si è "assestato" in modo 128, si digiti in modo diretto...
Sys 7134
... ed il gioco è fatto.

Un nuovo List confermerà che le due righe di programma sono state trasferite senza problemi, pronte per essere eventualmente rielaborate ed ancora trasferite a propria discrezione.

LIMITI E PRECAUZIONI

Qualora nel passaggio da C/64 a C/128 si omettesse di premere il tasto Restore prima del reset, non solo si perderebbe (come di norma) il programma in memoria, ma si renderebbe impossibile un normale ritorno al C/64, se non spegnendo prima l'apparecchio.

```
100 REM-----------------------------------------------------------------
110 REM          PASSAGGIO DA C/128 A C/64 (E VICEVERSA)
120 REM          CONSERVANDO IL LISTATO BASIC IN MEMORIA
130 REM-----------------------------------------------------------------
140 REM         ATTIVARE QUESTO PROGRAMMA IN MODO 128!!!
150 REM-----------------------------------------------------------------
160 :
170 FOR X=0 TO 249:READ A:POKE 32768+X,A:B=B+A:NEXT
180 IF B <> 32210 THEN PRINT "ERRORE NELLE RIGHE DATA!":END
190 POKE4626,255:POKE4627,127:REM --> ABBASSA TETTO DEL BASIC
200 END
210 DATA 009,128,009,128,195,194,205,056,048,032,163,253,032,080
220 DATA 253,032,021,253,032,091,255,088,032,083,228,032,191,227
230 DATA 032,034,228,160,000,169,128,132,051,132,055,133,052,133
240 DATA 056,169,101,141,000,128,141,002,128,173,016,018,133,045
250 DATA 173,017,018,233,020,133,046,132,251,132,253,169,028,133
260 DATA 252,169,008,133,254,200,177,251,145,253,200,208,249,165
270 DATA 254,197,046,240,006,230,252,230,254,208,237,032,051,165
280 DATA 076,116,164,120,032,188,246,032,225,255,208,003,076,102
290 DATA 254,162,000,134,251,134,253,165,046,133,252,105,019,133
300 DATA 254,141,255,027,164,045,140,254,027,177,251,145,253,136
310 DATA 192,255,208,247,165,252,201,008,240,006,198,252,198,254
320 DATA 208,235,169,009,141,000,128,141,002,128,162,000,189,195
330 DATA 128,157,222,027,232,201,096,208,245,032,129,255,162,000
340 DATA 189,226,128,201,000,240,254,032,210,255,232,208,243,162
350 DATA 000,169,128,142,018,018,141,019,018,173,254,027,141,016
360 DATA 018,173,255,027,141,017,018,169,029,141,002,028,032,079
370 DATA 079,096,079,082,065,032,080,082,069,077,073,032,073,076
380 DATA 032,083,079,076,079,032,082,069,083,069,084,000
390 END
```

In alternativa, per disabilitare la nostra routine, è sufficiente digitare Poke 32772, 0 (sia da C/64 che da C/128).

Inoltre, trovandosi in modo 64, occorre prestare una certa attenzione quando si utilizza Run / Stop + Restore.

Questa eventualità è prevista (ed ammessa) dal programma LM, ma si tenga ben premuto il tasto Run / Stop prima del Restore, altrimenti potrebbe attivarsi la routine di trasferimento.

In questo caso, comunque, nulla di drammatico: all'apparire del messaggio, basterà passare in modo 128 col reset, per poi tornare al 64.

Un altro elemento da prendere in considerazione, a proposito del modo 128, riguarda i comandi grafici.

Se, infatti, viene adoperata una istruzione Graphic per attivare l'alta risoluzione, l'inizio del Basic viene spostato a partire da $4000 (sempre di banco 0), rendendo inefficace il trasferimento attuato dalla nostra routine, qualora si decidesse di passare al modo 64.

Quindi, se in precedenza si fosse adoperata la pagina grafica, non si dimentichi di impartire...
Graphic Clr
...prima di Go64.

Infine, per i meno esperti, un'ultima (e scontata) avvertenza.

Non si dimentichi che un programma scritto in Basic 2.0 (quello del C/64) "gira" indifferentemente nei due computer, mentre i comandi specifici del Basic 7.0 (quello del C/128) non vengono riconosciuti in ambiente 64, tanto in fase di esecuzione che dopo un semplice List.

DA C/128 A C/64

Come già accennato, anche dopo un semplice Go64, un programma contina ad essere presente in memoria a partire da $1C00.

Il Basic del C/64, o meglio i suoi puntatori di inizio del testo (si veda la figura 2), richiedono però che, per essere riconosciuto dal sistema, debba risiederere da $800.

Le strade possibili, sono perciò due: trasferire le locazioni da $1300 in poi nell'area che inizia da $800, o alterare i puntatori di inizio Basic, in modo che "mirino" a $1C00.

Il problema, quale che sia la scelta da adottare, è che tali operazioni devono essere svolte quando già ci si trova in modo 64.

Per fortuna ci viene incontro la routine di reset del C/64, cui il C/128 cede il controllo dopo il Go64.

Questa routine, com'è noto alla vecchia guardia sessantaquattrista, esegue prima di ogni altra cosa un controllo sulla eventuale presenza di cartridge esterne, presenza che viene rivelata dai codici ascii "CBM80" locati nei 5 byte da $8004 (32772)in poi.

Se trova questi caratteri (CBM80), il sistema salta all'indirizzo contenuto in $8000 ed $8001 (in formato basso/alto, come sempre).

Ebbene, quando lanciamo il listato caricatore dal 128, questo per prima cosa alloca i caratteri CBM80 al giusto posto, preceduti dall'indirizzo di inizio della nostra routine.

Inoltre, sempre quando ancora ci si trova in modo 128, il tetto del Basic viene "abbassato" (riga 190 del listato) per evitare interferenze in questa area di memoria.

Passando al 64, come ormai dovrebbe essere chiaro, le locazioni da $8000 in poi vengono mantenute integre, per cui, quando il reset di sistema del 64 è chiamato in causa, crederà di avere a che fare con una cartuccia esterna, e cederà il controllo alla routine puntata da $8000-$8001 e dalle due locazioni che seguono.

Praticamente verrà attivato il nostro programma LM, il cui disassemblato è pubblicato in queste stesse pagine (in formato simbolico MacroAssembler).

L'inizio vero e proprio, contrassegnato dalla label Res64, deve sobbarcarsi il compito di ultimare le procedure appena iniziate dal nuovo s.o.

Infatti, dato che il controllo su eventuali cartridge è la prima fase del reset, al momento in cui esso viene dirottato, il sistema non è ancora inizializzato, e quindi non in grado di funzionare correttamente.

Ecco dunque il motivo delle righe 8-15, copia integrale delle istruzioni presenti nella ROM del C/64 da $FCF2 a $FCFE e da $E394 a $E39A.

Sbrigata questa formalità, la routine continua abbassando anche qui il tetto del Basic e la memoria riservata alle stringhe (17-22) per gli stessi motivi già visti a proposito del C/128; in seguito modifica gli indirizzi utili al reset (24-26).

Con quest'ultima operazione, in pratica, si prepara il terreno al passaggio inverso (da C/64 a C/128).

```
001            *=$8000        Start a 32768.
002  ;-----------------------------------
003  ADDR1     .WOR RES64     Dirottamento del
004  ADDR2     .WOR RES64     reset del C/64.
005            .BYT $C3,$C2   Caratteri ascii
006            .BYT $CD,'80'   "CBM80".
007  ;-----------------------------------
008  RES64     JSR $FDA3      Copia della
009            JSR $FD50      routine di reset
010            JSR $FD15      del C/64, con
011            JSR $FF5B      esclusione del
012            CLI            controllo sulla
013            JSR $E453      presenza di
014            JSR $E3BF      eventuali car-
015            JSR $E422      tucce esterne.
016  ;-----------------------------------
017            LDY #$00       Si modificano i
018            LDA #$80       puntatori a
019            STY $33        tetto basic e
020            STY $37        inizio delle
021            STA $34        stringhe (nuovo
022            STA $38        tetto=$8000).
023  ;-----------------------------------
024            LDA #<GO128    Nuovo indirizzo
025            STA ADDR1      per reset (o
026            STA ADDR2      altro NMI).
027  ;-----------------------------------
028            LDA $1210      Byte basso di
029            STA $2D        fine programma.
030            LDA $1211      Byte alto dello
031            SBC #$14       stesso, meno 20
032            STA $2E        per il C64.
033  ;-----------------------------------
034            STY $FB        Setta puntatori
035            STY $FD        FB-FC (partenza)
036            LDA #$1C       con $1C00, e
037            STA $FC        puntatori FD-FE
038            LDA #$08       (arrivo) con
039            STA $FE        valore $0800.
040  ;-----------------------------------
041            INY            Indice Y = 1.
042  LOOP      LDA ($FB),Y    Trasferisce area
043            STA ($FD),Y    da $1C01 (basic
044            INY            C128) a $0801
045            BNE LOOP       (basic C64).
046  ;-----------------------------------
047            LDA $FE        Controlla se
048            CMP $2E        fine programma.
049            BEQ CONT       Se si', salta...
050            INC $FC        altrimenti ag-
051            INC $FE        giorna puntatori
052            BNE LOOP       e ricomincia.
053  ;-----------------------------------
054  CONT      JSR $A533      Link del basic.
055            JMP $A474      Salta a Ready.
056  ;===================================
057  GO128     SEI            Disabilita IRQ.
058            JSR $F6BC      Scansione tasti.
059            JSR $FFE1      Stop premuto?
060            BNE CONT2      Se no, salta.
061            JMP $FE66      Ritorno al basic.
062  ;-----------------------------------
063  CONT2     LDX #$00       Setta puntatori
064            STX $FB        FB-FC con indi-
065            STX $FD        rizzo fine prg.
066            LDA $2E        del C64 e FD-FE
067            STA $FC        con stesso indi-
068            ADC #$13       rizzo, ma +19
069            STA $FE        nel byte alto.
070            STA $1BFF      HiByte per C128.
071  ;-----------------------------------
072            LDY $2D        Low byte in
073            STY $1BFE      buffer per C128.
074  LOOP2     LDA ($FB),Y    Trasferisce area
075            STA ($FD),Y    da fine prg C64
076            DEY            a fine prg C128
077            CPY #$FF       procedendo a
078            BNE LOOP2      ritroso.
079  ;-----------------------------------
080            LDA $FC        Se giunti a $800
081            CMP #$08       (inizio prg.),
082            BEQ FINE       salta a riga 87.
083            DEC $FC        Se no, aggiorna
084            DEC $FE        puntatori (hi)
085            BNE LOOP2      e continua.
086  ;-----------------------------------
087  FINE      LDA #<RES64    Ripristina i
088            STA ADDR1      puntatori del
089            STA ADDR2      reset.
090            LDX #$00       Copia nelle
091  LOOP3     LDA ROUT2,X    locazioni da
092            STA $1BDE,X    $1BDE in poi
093            INX            (buffer per 128)
094            CMP #$60       la routine di
095            BNE LOOP3      riga 105.
096            JSR 65409      Pulisce schermo.
097            LDX #$00       Preleva caratte-
098  LOAD      LDA MESSG,X    re di riga 119.
099            CMP #$00       Se car.=0, si
100  STOP      BEQ STOP       blocca in attesa.
101            JSR $FFD2      Stampa carattere.
102            INX            Incrementa indice
103            BNE LOAD       e continua.
104  ;===================================
105  ROUT2     LDX #$00       Tetto per testo
106            LDA #$80       del basic in
107            STX $1212      bank 0 del C128
108            STA $1213      ($8000).
109  ;-----------------------------------
110            LDA $1BFE      Preleva dal buf-
111            STA $1210      fer indirizzo
112            LDA $1BFF      di fine prg e lo
113            STA $1211      usa per puntatori.
114            LDA #$1D       Valore casuale
115            STA $1C02      per link basic.
116            JSR $4F4F      Relink del prg.
117            RTS            Return.
118  ;===================================
119  MESSG     .BYT 'ORA PREMI'   Caratteri
120            .BYT ' IL SOLO'    messaggio
121            .BYT ' RESET',0    per C64.
122            .END           Fine routine LM.
```

Poco dopo...

Pronto, AEGIS, avremmo bisogno di un favore: ci tirate per 64 blocchi?

La risposta è negativa...

Giovedini, ci dispiace ma non possiamo...

Qualcuno ci ha messo in PAUSA e stiamo aspettando di uscirne!

GAME IN PAUSE
PRESS FIRE TO CONTINUE

Infatti, rimanendo inalterati i caratteri CBM80 di cui sopra, in caso di reset (o, come vedremo tra breve, di altro Interrupt non mascherabile), verrà attivata la sezione della routine specificata da Go128 (riga 57).

E, finalmente, si passa al trasferimento del programma Basic vero e proprio (34-52).

Si è scelta tale soluzione, piuttosto che la modifica dei puntatori di inizio-Basic, per non limitare ulteriormente la memoria a disposizione, già decurtata delle locazioni da $8000 ad $A000.

La tecnica di trasferimento è incentrata sull'uso dei puntatori FB-FC e FD-FE, funzionali all'indirizzamento indiretto delle righe 42 e 43.

Si noti come l'indirizzo di fine del programma Basic è prelevato direttamente dalle locazioni $1210 e $1211, che altro non sono se non i puntatori di fine-Basic del C/128.

Questi, per nostra fortuna, vengono anch'essi conservati dopo il passaggio al modo 64, per cui, con una semplice sottrazione a carico del byte alto (30-32), è possibile aggiornare lo stesso tipo di puntatori, ma del C/64 (si veda ancora la figura 2).

A trasferimento ultimato, basta un salto alla routine di Link ($A533) per "ricollegare" correttamente le righe del Basic, e tutto è pronto per il Ready (riga 55).

DA C/64 A C/128

Per il passaggio inverso, non si dispone degli stessi automatismi offerti dalla particolare gestione del reset del C/64, per cui è necessario arrangiarsi da soli, aggiungendo qualche "manovra" in più alla procedura.

Continuiamo, dunque, a procedere di pari passo con il disassemblato, e tutto risulterà (forse) più chiaro.

La sezione da riga 57 in poi, viene attivata dalla pressione del tasto Restore, che genera nel sistema un cosiddetto Interrupt non mascherabile (NMI), gestito in Rom (sempre del C/64) da una routine allocata a partire da FE43.

Anche questa, come il reset, effettua lo stesso tipo di controllo su eventuali cartucce esterne, per cui, dopo il nostro dirottamento, salterà... dove vogliamo noi (a riga 57, in questo caso).

Normalmente la stessa routine esegue un controllo sul tasto Run/Stop, ma, dopo la nostra manipolazione, anche in questo caso dovremo sostituirci al sistema (righe 57-61) per rendere possibile l'uso di Run/Stop + Restore.

In caso contrario, tale funzione sortirebbe come unico effetto quello di attivare la nostra routine.

Con le righe da 63 a 85, si effettua un trasferimento opposto a quello già visto, utilizzando gli stessi puntatori, ma procedendo dalla fine del programma verso l'inizio, per evitare indesiderate sovrapposizioni.

Inoltre l'indirizzo di fine programma viene memorizzato nelle locazioni $1BFE-$1BFF, in modo da essere recuperato quando ci si troverà in modo 128 (si tenga presente lo schema di figura 1!).

Ultimata questa fase, vengono ripristinati al valore iniziale gli indirizzi che serviranno per tornare in modo 64 (87-89), e si copia nell'area comune (usata come buffer) da $1BDE la routine che inizia da riga 105.

Questa, in pratica, dopo il passaggio al modo 128, resterà immutata nelle locazioni libere da interferenze del Basic.

A questo punto (96-103) viene visualizzato il messaggio presente da riga 119 in poi, e si crea un loop infinito (100) in attesa che venga premuto il reset.

Finalmente, tornati al 128, non rimane che attivare, con Sys 7134, la routine di riga 105, che inizierà con il solito abbassamento del tetto del Basic (105-1080).

Verrà poi prelevato l'indirizzo di fine Basic e schiaffato nei suoi naturali puntatori (110-113), e, dopo un salto alla routine di Link del 128 ($4F4F di banco 15), tutto si concluderà (era ora!) con il solito rassicurante ritorno al cursore lampeggiante.

A dispetto della lunga, e forse estenuante, descrizione, il tutto avviene in termini di tempo assolutamente accettabili, come d'altronde garantito dal linguaggio macchina.

Qualcosa in più, di certo, occorrerà a voi per "assimilare" quanto finora descritto... ma non si può certo competere con un ammasso di chip!

IMMAGINI: PICCOLE, PICCOLISSIME, GRANDI, GRANDISSIME

Se amate la grafica hi-res (e se, in particolare, possedete il programma "Print Shop 64") potrete manipolare immagini a volontà

di **Tito Brasolin**

Operando con una schermata in alta risoluzione può talvolta risultare utile salvare su disco solo una parte della medesima, per poi richiamarla, ed eventualmente inserirla, in altre pagine grafiche di propria realizzazione.

Il programma proposto serve appunto a svolgere tale (gravoso) compito. Inoltre è in grado di moltiplicare le dimensioni di una figura a mano a mano che la carica da disco, e presenta una certa compatibilità con alcuni programmi commercializzati.

E' bene precisare fin da ora che la routine non salva nè carica i dati relativi al colore ed è quindi necessario ricorrere a qualche artifizio, del resto semplicissimo, per colorare le immagini.

Ma veniamo alla sintassi della routine:

per il salvataggio:

Sys xxxx, mode, pg, n$, x0, y0, xl, yl

per il caricamento:

Sys xxxx, mode, pg, n$, x0, y0, fx, fy [, mask [, xl [, yl]]]

Il parametro XXXX è, ovviamente, l'indirizzo di partenza della routine, interamente rilocabile (attenzione, però, a non combinare pasticci sovrapponendo la routine a pagine grafiche!).

Il parametro MODE può assumere un qualsiasi valore compreso tra zero e 255, ma permette, in realtà, di scegliere tre sole alternative:

mode = 0: seleziona il caricamento "normale". La finestra, in fase di ingrandimento, viene trattata come una immagine in bit map.

mode = 1: seleziona il caricamento in modo multicolore. In questo modo, durante la fase di ingrandimento, saranno rispettate le coppie di bit affiancati che determinano il colore nel modo multicolor.

Per chiarire il significato dei due diversi modi di ingrandimento, consideriamo un esempio.

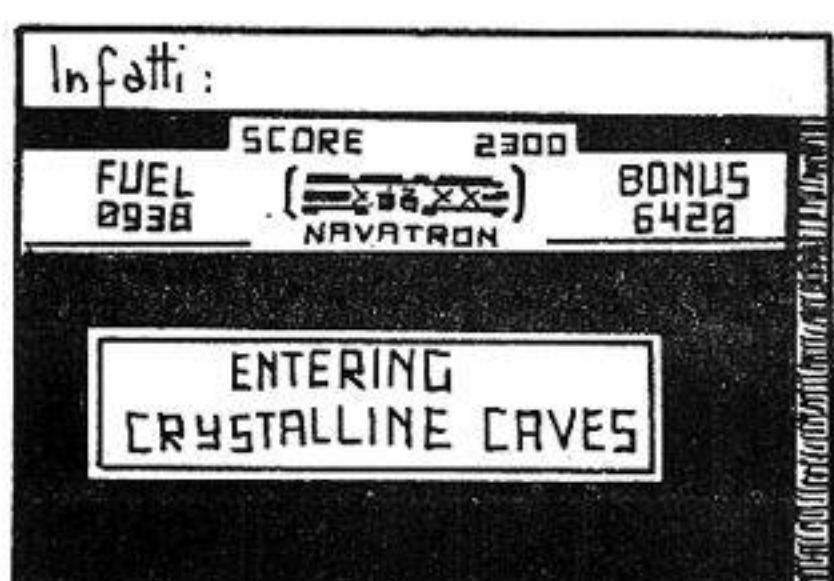
Poniamo di dover raddoppiare le dimensioni orizzontali di un byte settato al valore binario 01011001.

Ingrandendolo in modo bit map, otterremmo i due bytes, affiancati...

00110011
11000011

Invece, in modo multicolor, otterremmo:
01010101
10100101

Il modo migliore di apprezzare la differenza è di sperimentare la routine.

Prima di usare il programma, realizzate un'immagine hi-res che funga da "cavia"

```
100 REM
110 REM PROGRAMMA MANIPOLATORE DI IMMAGINI
120 REM
130 REM ROUTINE IN LINGUAGGIO MACCHINA INTERAMENTE RILOCABILE.
150 REM PER UTILIZZARLA, LEGGI LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
175 X=49152:REM DEFAULT
180 Y=-1:GOSUB 1000:PRINT:INPUT"DA QUALE LOCAZIONE";X
190 READW:Y=Y+1:IFW<0THEN210
200 GOTO190
210 PRINT"PRIMA LOCAZIONE ="X
220 PRINT"ULTIMA LOCAZIONE ="X+Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GETA$:IFA$=""THEN240
250 IFA$="S"THENRESTORE:T=X:GOTO280
260 RUN
270 :
280 GOSUB1000:PRINT:PRINT"ATTENDERE...":PRINT:W=0
290 READB:IFB>=0THENPOKET,B:T=T+1:W=W+B:GOTO290
300 READB:IFB<>WTHENPRINT:PRINTCHR$(18)"ERRORE DI TRASCRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X"A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PRINT
330 X1=INT(X/256):X2=X-(X1*256):PRINT"POKE44,"X1":POKE43,"X2;
340 Y1=INT(T/256):Y2=T-(Y1*256):PRINT":POKE46,"Y1":POKE45,"Y2:END
1000 PRINTCHR$(147);
1010 PRINT"ICON MANAGER":PRINT
1020 PRINT"SALVATAGGIO: SYS XXXX,MODE,PG,N$,X0,Y0,   XL,YL"
1030 PRINT"CARICAMENTO: SYS XXXX,MODE,PG,N$,X0,Y0,   FX,FY[,MASK[,XL[,YL]]]"
1040 PRINT
1050 PRINT"XXXX :INDIRIZZO DI PARTENZA QUALSIASI"
1060 PRINT"MODE :SELEZIONA SALVATAGGIO (>1)"
1070 PRINT"      O CARICAMENTO NORMALE (0) O IN          M.C.M (1)"
1080 PRINT"PAGE :BYTE ALTO INIZIO PAGINA GRAFICA"
1090 PRINT"N$   :NOME FILE"
```

Solo il demo pubblicato necessita di "Print Shop"

La terza alternativa fornita dal parametro MODE è quella del salvataggio, ottenuta con un qualsiasi valore di mode maggiore di 1 e minore di 256. Non è prevista (e comunque sarebbe superflua) la possibilità di variare le dimensioni di una finestra durante il salvataggio, o di distinguere, anche in questo caso, fra bit map e multicolor.

Il parametro PG è semplicemente il byte alto della locazione iniziale della pagina in alta risoluzione con cui stiamo operando. Se, ad esempio, la pagina inizia a 8192, risulta...

PG = 8192 / 256 = 32

Il nome da assegnare al file in salvataggio (o da caricare) è contenuto in N$, variabile stringa oppure stringa "esplicita".

Si fa notare che i files generati dalla routine sono di tipo programma (prg).

I parametri X0 e Y0 sono le coordinate dell'angolo superiore sinistro della finestra con cui stiamo operando.

L'ascissa X0 può variare fra 0 e 39 e l'ordinata Y0 fra 0 e 199. Le coordinate (0, 0) corrispondono all'angolo superiore sinistro dello schermo. Per ragioni di comodità, mentre l'ordinata può variare bit per bit, l'ascissa varia byte per byte, più o meno come se ci trovassimo in una normale pagina di testo.

Da questo momento in poi sono presenti sostanziali differenze fra i parametri da fornire per il caricamento e quelli per il salvataggio. Trattiamo dunque separatamente i due aspetti.

```
1100 PRINT"X0    :ASCISSA ANGOLO SUPERIORE SIN.DELLA      FINESTRA (0...39)"
1110 PRINT"Y0    :ORDINATA ANGOLO SUP. SIN. DELLA         FINESTRA (0...199)"
1120 PRINT"XL,YL:LARGHEZZA E ALTEZZA FINESTRA            (0...255 CON 0=256)"
1130 PRINT"FX,FY:FATTORI INGRANDIMENTO ORIZZ. E          VERT. (0...255,0=256)"
1140 PRINT"MASK :MASCHERA PER OR ESCLUSICO               DEFAULT=0"
1150 RETURN
1160 DATA 032,241,183,142,255,003,160,000,224,002,144,002,160,001,169,002
1170 DATA 162,008,032,186,255,032,241,183,142,253,003,032,253,174,032,158
1180 DATA 173,032,130,183,152,166,034,164,035,032,189,255,032,241,183,224
1190 DATA 040,144,005,162,014,108,000,003,142,052,003,032,241,183,224,200
1200 DATA 176,241,134,002,032,241,183,142,054,003,142,057,003,142,058,003
1210 DATA 032,241,183,142,056,003,142,059,003,164,185,240,027,032,192,255
1220 DATA 162,002,032,201,255,173,054,003,032,210,255,173,056,003,032,210
1230 DATA 255,160,000,140,254,003,240,061,032,192,255,162,002,032,198,255
1240 DATA 032,207,255,141,054,003,032,207,255,141,056,003,162,000,032,121
1250 DATA 000,201,044,208,003,032,241,183,142,252,003,032,121,000,201,044
1260 DATA 208,019,032,241,183,142,054,003,032,121,000,201,044,208,006,032
1270 DATA 241,183,142,056,003,173,054,003,141,055,003,173,052,003,133,253
1280 DATA 165,144,240,008,032,204,255,169,002,076,195,255,165,185,041,159
1290 DATA 208,090,032,207,255,077,252,003,141,019,003,172,057,003,140,254
1300 DATA 003,169,001,141,053,003,174,058,003,173,019,003,172,255,003,010
1310 DATA 046,053,003,136,016,249,202,208,006,174,057,003,141,019,003,144
1320 DATA 232,142,058,003,174,059,003,206,254,003,024,144,031,173,053,003
1330 DATA 145,251,202,208,027,165,254,230,253,172,254,003,208,195,206,055
1340 DATA 003,208,157,133,002,206,056,003,208,139,240,152,165,002,133,254
1350 DATA 165,254,201,200,144,004,233,200,133,254,041,248,133,251,074,074
1360 DATA 074,109,253,003,133,252,165,253,201,040,144,005,233,040,133,253
1370 DATA 024,101,251,133,251,169,000,006,251,042,006,251,042,006,251,042
1380 DATA 101,252,133,252,165,254,041,007,168,230,254,165,185,041,159,240
1390 DATA 156,173,014,220,170,041,254,141,014,220,165,001,041,248,133,001
1400 DATA 177,251,168,165,001,009,007,133,001,142,014,220,152,032,210,255
1410 DATA 024,144,130
1420 DATA -1,48868
1430 END
```

SALVATAGGIO

Il salvataggio avviene da sinistra a destra e dall'alto verso il basso.

I parametri XL e YL sono, rispettivamente, larghezza (in bytes) e altezza (in bits) della finestra da salvare. Essi sono anche i primi dati che la routine riversa su disco, al posto dell'indirizzo di partenza che, in genere, caratterizza i files di tipo PRG. Sarebbe un grave errore, quindi, caricare tali file con il suffisso ",8,1".

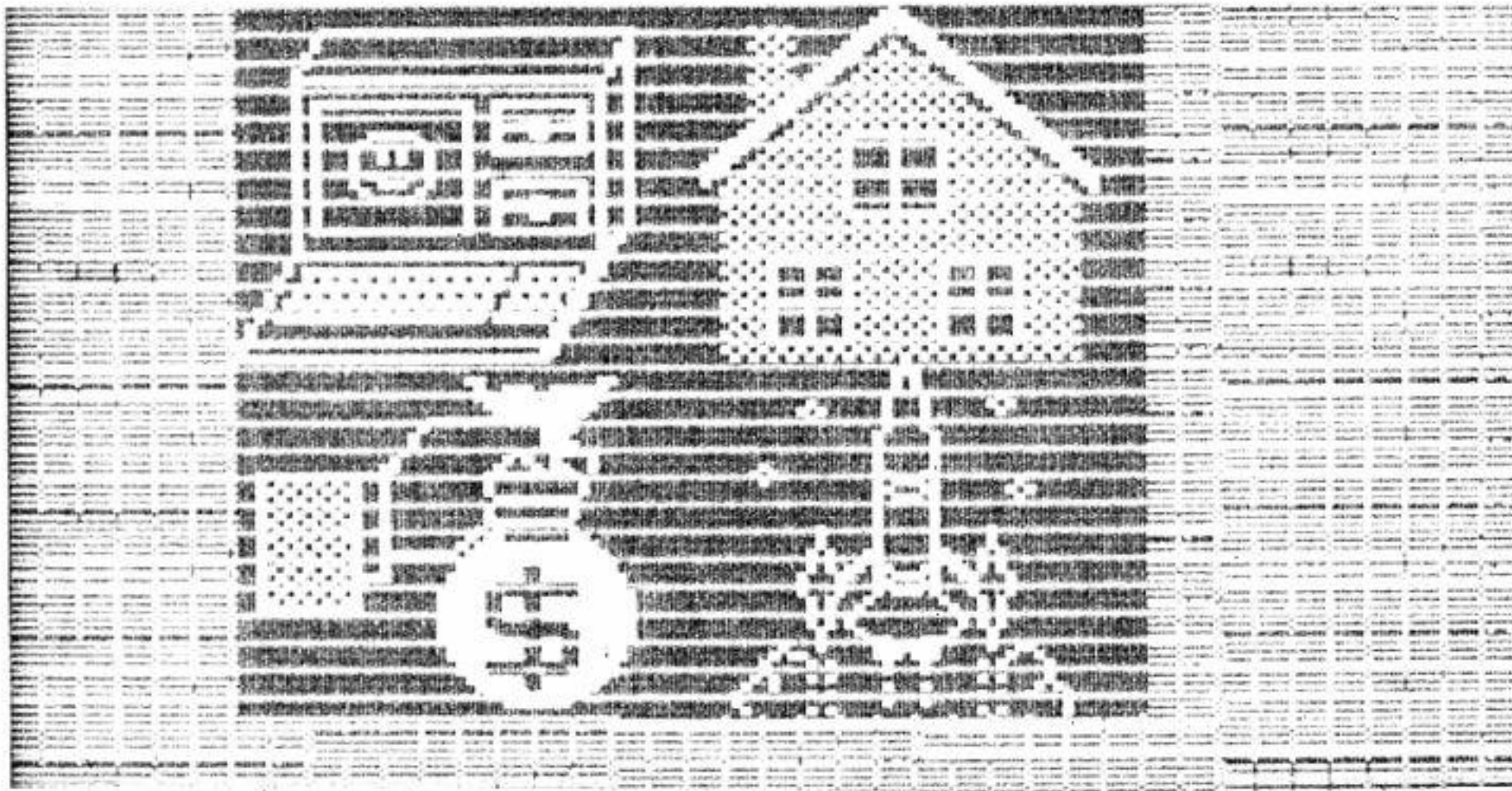
In teoria possono assumere qualunque valore compreso fra zero e 255, con lo zero che vale come 256. Tuttavia la routine è strutturata in modo tale che quando, nella fase di salvataggio, si raggiunge il bordo destro, si riporta al sinistro e prosegue da lì. Analogamente, giunta al fondo, riparte dall'alto. Per questo motivo larghezze XL superiori a 40 e altezze YL superiori a 200 causeranno un doppio salvataggio di alcuni dati.

Tale particolare struttura della routine si rivela utile quando si vuole salvare, in un unico file, più finestre poste a lati opposti o addirittura ai quattro angoli dello schermo. Ad esempio, con un'istruzione del tipo...

Sys xxxx, 2, pg, n$, 35, 160, 10, 80

...si salvano assieme quattro immagini di formato 5 bytes per 40 bits poste ai quattro angoli della pagina grafica in questione. Potremo, in seguito, caricarle nello stesso punto o in una zona diversa dello schermo.

La routine è interamente rilocabile: provate a "fonderla" con una di hard copy

```
100 REM * ICON MANAGER
110 REM * BY TITO BRASOLIN
120 REM * GIUGNO 1989
130 REM * DIMOSTRAZIONE CON
140 REM * ROUTINE ALLOCATA DA 49152
145 REM INDISPENSABILE PROGRAMMA PRINT SHOP!!!
150 PRINT"1 LEGGO DA PRINT SHOP"
155 PRINT"2 CARICO DEMO":INPUT X:IF X=2 THEN 410
160 POKE 55,0: POKE 56,32
170 S$=" |||||||||" : REM COMMODORE+M
180 PRINT CHR$(147);
190 PRINT "INSERISCI UNA LIBRARY DEL PRINT SHOP"
200 PRINT "DIMMI IL NOME DI QUATTRO ICONE":PRINT
210 FOR R=0 TO 3
220 PRINT R+1;:INPUT N$(R)
230 REM CONTROLLA PRESENZA FILE
240 OPEN 2,8,0,N$(R)
250 OPEN 15,8,15:INPUT#15,A,B$
260 PRINT A,B$: CLOSE2:CLOSE15
270 IF A<>0 THEN 220
280 NEXT R: PRINT CHR$(147);
290 GOSUB 460
300 FOR X=0 TO 1
310 FOR Y=0 TO 1
320 SYS 49152,0,32,N$(Y+2*X),X*34,Y*155,1,1,0,6,45
330 NEXT Y
340 NEXT X
350 POKE198,0:WAIT198,1
360 SYS 58792
370 PRINT CHR$(147);"INSERISCI UN DISCHETTO FORMATTATO E CON UN PO' DI SPAZIO"
380 PRINT "POI PREMI UN TASTO"
390 POKE 198,0:WAIT 198,1
400 SYS 49152,2,32,"DIMOSTRAZIONE",34,155,12,90
410 GOSUB 460
420 SYS 49152,0,32,"DIMOSTRAZIONE",6,0,2,2,255
430 POKE 198,0:WAIT 198,1
440 SYS 58792
450 END
460 S$=" |||||||||" : REM COMMODORE+M
465 FOR K=0 TO 110: PRINT S$;:NEXT K:POKE 2023,103
470 POKE 53265,PEEK(53265)OR32
480 POKE 53272,PEEK(53272)OR8
490 RETURN
```

CARICAMENTO

Come già accennato, la routine consente di ingrandire una finestra in caricamento tramite i parametri FX e FY, fattori di ingrandimento, rispettivamente, orizzontale e verticale. Entrambi sono interi, variabili tra zero e 255, con lo zero che ha l'effetto di un 256; tuttavia solo i valori inferori saranno utili, compatibilmente con le dimensioni della finestra. Infatti, specie se questa non è molto piccola, con elevati valori di fx e fy rischia di ingrandirla oltre il bordo destro e inferiore e, analogamente a quanto visto per il salvataggio, potrebbe "sbucare" fuori da sinistra o dall'alto, sporcando la pagina grafica già disegnata.

Durante il caricamento la routine effettua un OR Esclusivo tra i bytes caricati e il valore del parametro MASK, intero compreso fra zero (valore di default) e 255. Ponendo mask = 255 l'immagine verrà caricata in negativo.

Gli ultimi due parametri sono XL e YL, dimensioni della finestra da caricare. Normalmente non servono, dato che i primi due bytes di ciascun file generato dalla routine contengono l'informazione.

Grazie ai due parametri, però, è possibile caricare e visualizzare anche icone tratte da library per i programmi "Print Shop" e "Print Shop Companion", il cui formato è identico al nostro, tranne appunto per quanto riguarda i primi due dati del file che non rappresentano le dimensioni della figura.

Per "Print Shop" dovremo porre XL=6 e YL=45, mentre per "Print Shop Companion" XL=11 e YL=52. Senza specificare tali valori otterremmo immagini senza senso. Ricordiamo che E' COMUNQUE IMPOSSIBILE caricare con i programmi citati files generati dalla nostra routine.

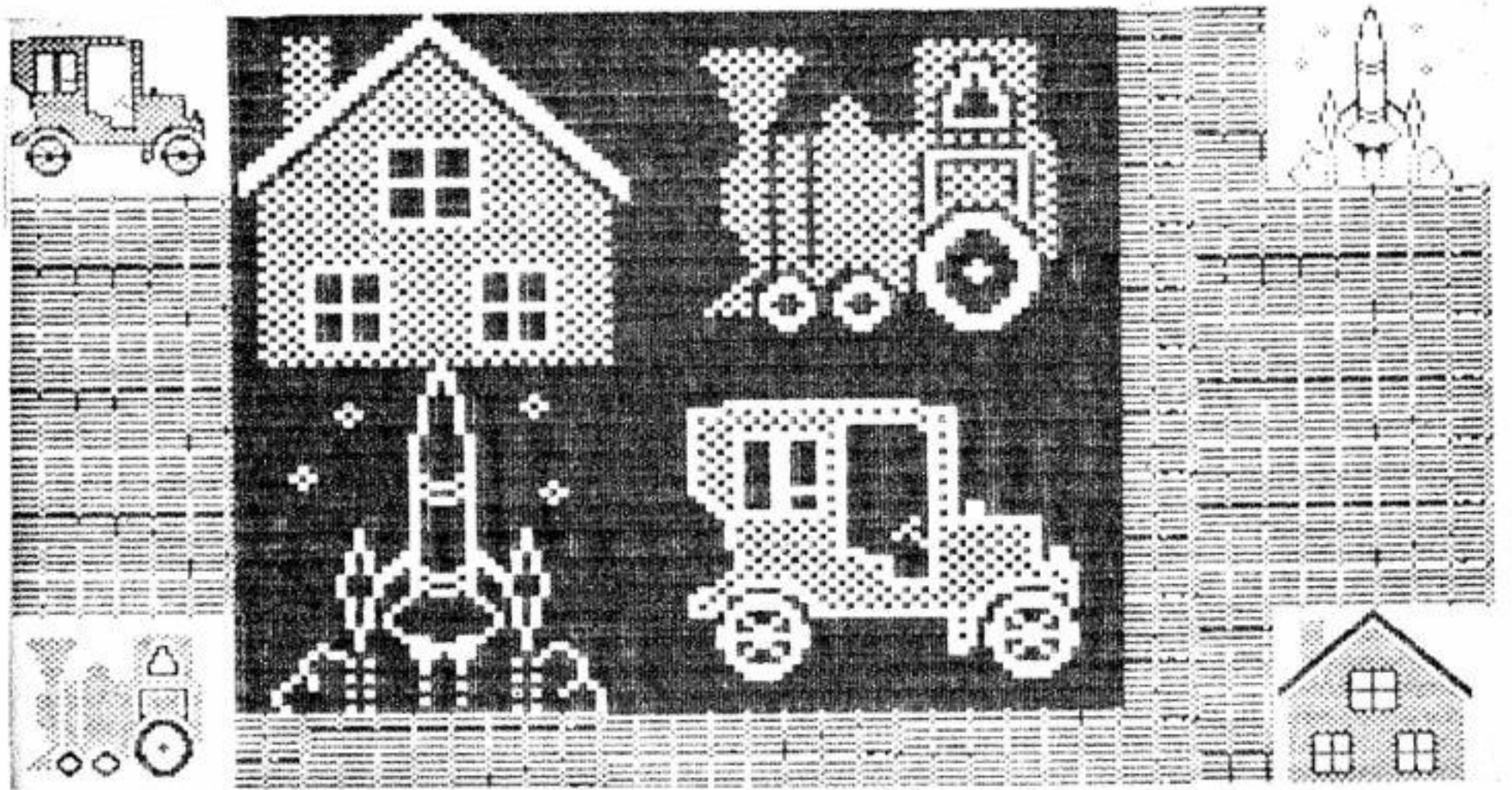
PER I PIU' BRAVI

Un breve cenno ad alcuni aspetti della routine che possono non risultare chiari dal disassemblato.

E' ovvio che un byte ingrandito FX volte si trasformerà in FX bytes. Un contatore alla rovescia, che parte dal valore fx, quando raggiunge lo zero avverte che è terminato l'ingrandimento orizzontale di un byte.

L'ingrandimento avviene facendo scivolare da destra a sinistra i bit del byte da ingrandire in una locazione destinata ad accogliere il byte che poi andrà stampato fy volte nella pagina grafica per completare l'ingrandimento verticale.

L'indice X funge da contatore per far sì che un bit (acceso o spento che sia) divenga fx bits una volta ingrandito.

Prima di visualizzare la pagina grafica, provvedete a colorarne lo sfondo

E' stata sostituita la familiare routine di plot...

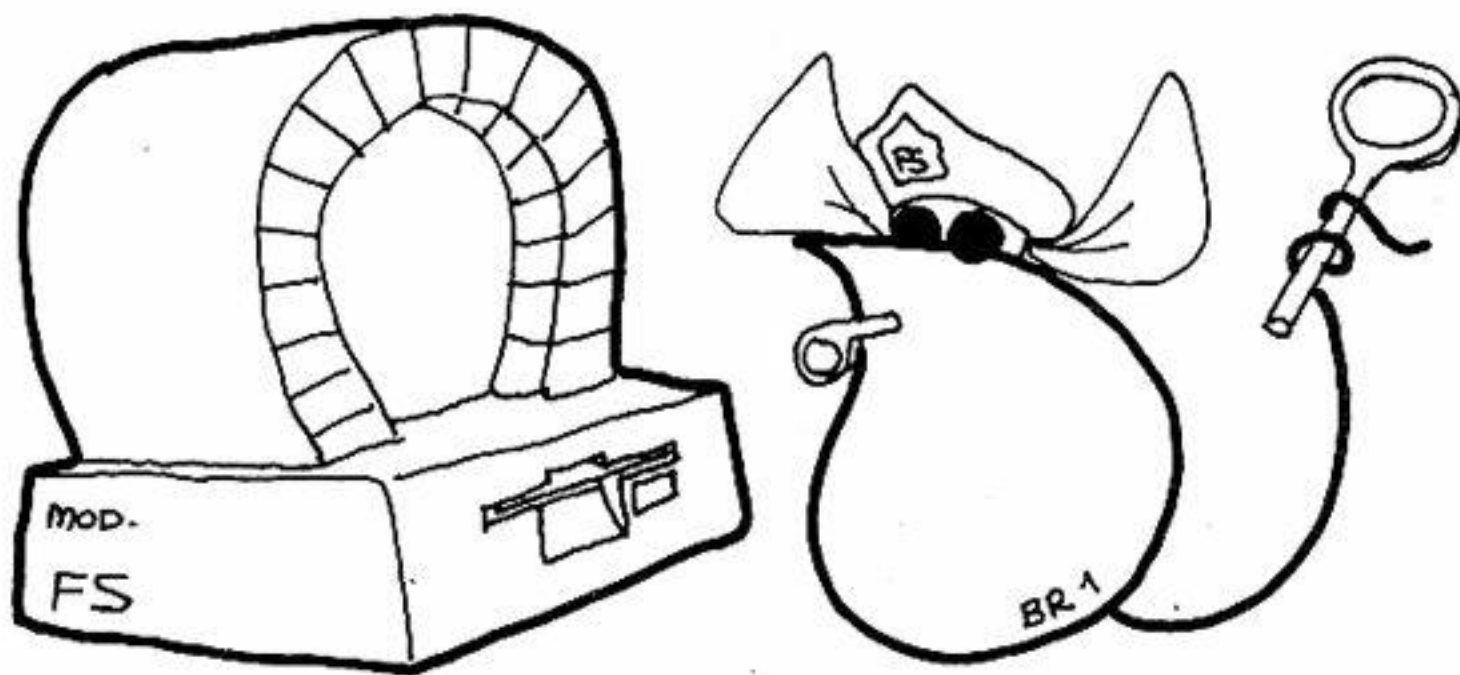
loc=page + (8*x) + 40* (y and 248) + (y and 7)

...con quella equivalente, decisamente più comoda...

loc = page + 8*(x+ (y and 248)) + 32*(y and 248) + (y and 7)

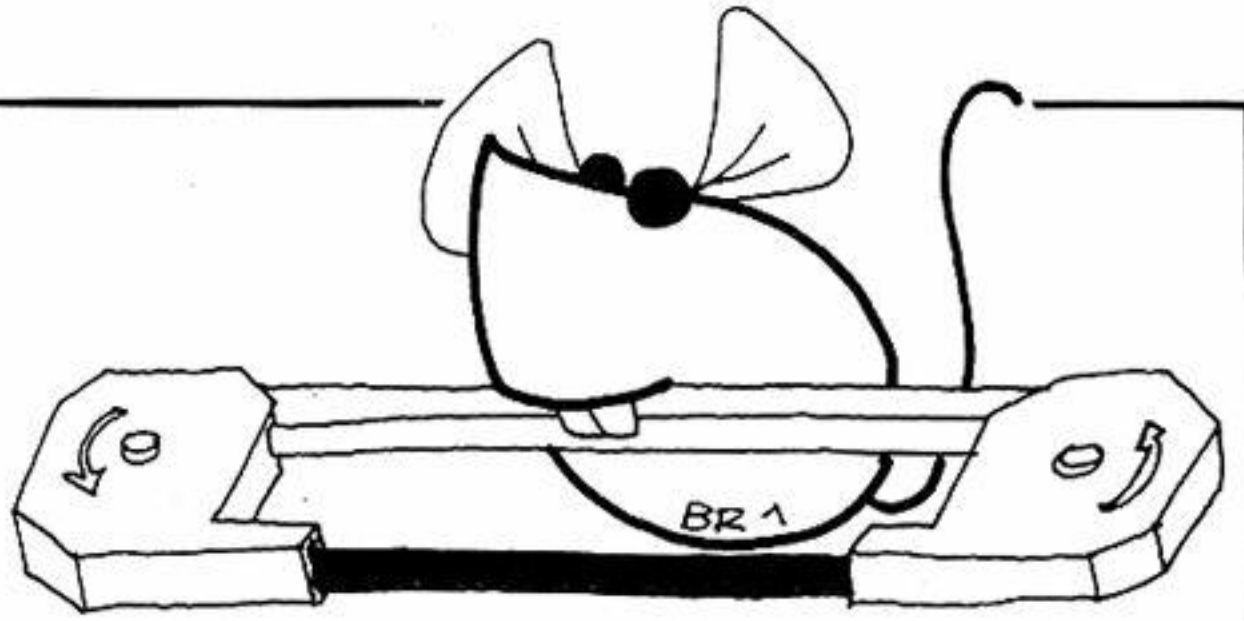
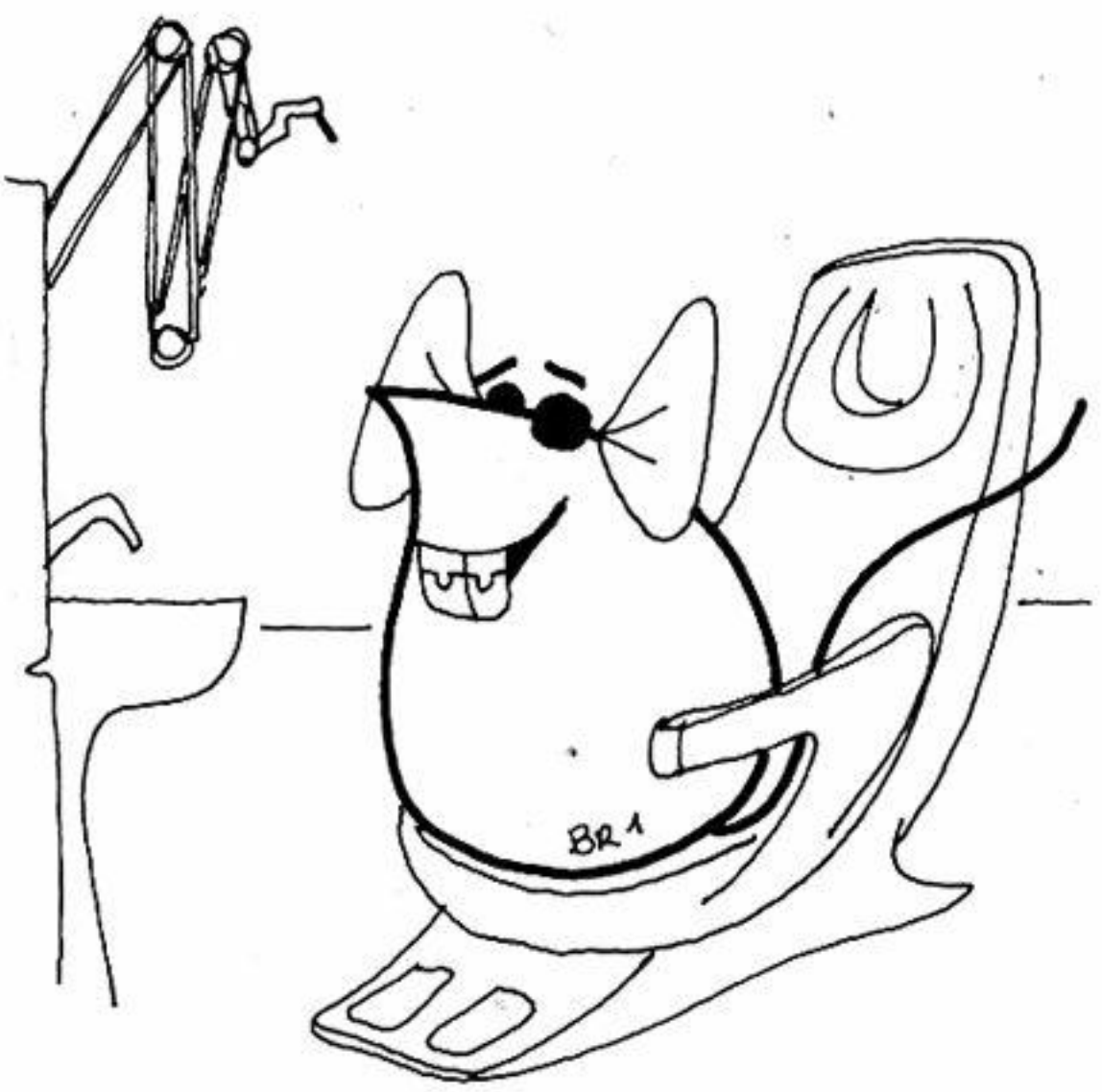
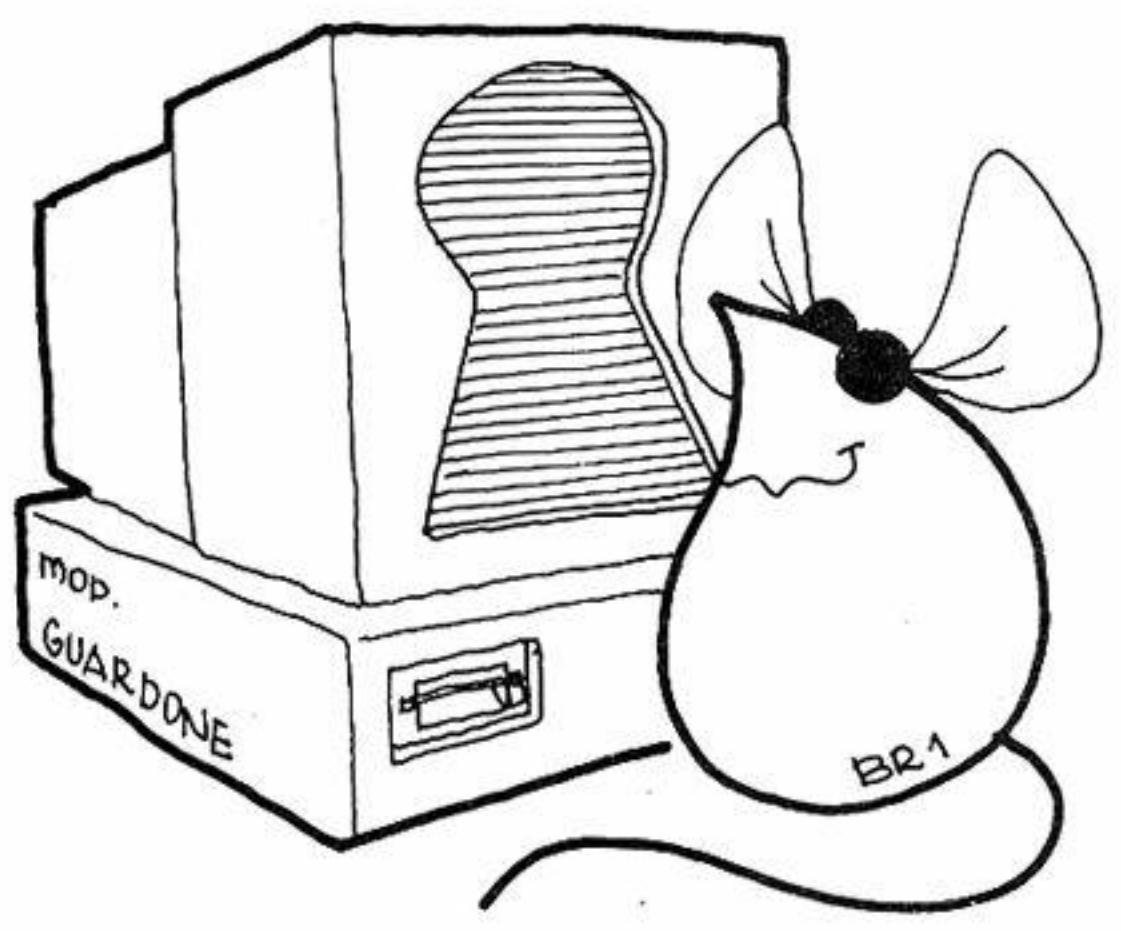
La routine è costituita da un blocco comune e da parti specifiche del caricamento o del salvataggio.

Quando è necessario verificare se si sta salvando, o caricando, si esamina il contenuto della locazione 185, che contiene l'attuale indirizzo secondario. E' necessario però effettuare un AND 159 poichè il S.O. esegue un OR 96 con il valore contenuto nella locazione.

Non esiste controllo di errore, ma una semplice lettura della variabile di stato: il programma cessa se questa è diversa da zero (fine del file). Pertanto è meglio effettuare alcuni controlli PRIMA di lanciare la routine, per sincerarsi che il file cercato esista o che non vi sia già su disco un file omonimo di quello che si sta per salvare.

IL DEMO

Il semplicissimo demo si limita a chiedere il nome di quattro icone di Print Shop da caricare, e ad evidenziarle agli angoli dello schermo. Attende quindi la pressione di un tasto e richiede l'inserimento di un disco formattato e libero. Salva poi le quattro figure come un file unico, con il metodo descritto in precedenza. Di seguito ricarica lo stesso file (che ha il nome standard di "dimostrazione") in negativo, di dimensioni raddoppiate e al centro (o quasi) dello schermo. Un'altra pressione di tasto e il programma termina.

```
;*************************************
;         PHOTO MANAGER 64
;
;         BY TITO BRASOLIN
;
;           GIUGNO 1989
;
;      DISASSEMBLATO COMMENTATO
;
;*************************************
          ;VARIABILI
CHKCOM    = $AEFD
FRMEVL    = $AD9E
LEN1      = $B782
SETLFS    = $FFBA
SETNAM    = $FFBD
OPEN      = $FFC0
CLOSE     = $FFC3
CHKIN     = $FFC6
CHKOUT    = $FFC9
CLRCHN    = $FFCC
CHRIN     = $FFCF
CHROUT    = $FFD2
CHRGOT    = $79
          ;
SA        = $B9
YØ        = $02
ADRS      = $FB
X         = $FD
Y         = $FE
BYTE      = $0313
XØ        = $0334
BY        = $0335
XL        = $0336
CONTXL    = $0337
YL        = $0338
FXØ       = $0339
FX        = $033A
FYØ       = $033B
MASK      = $03FC
PAGE      = $03FD
CONTFX    = $03FE
MULTIC    = $03FF
;*************************************
          JSR $B7F1      ;LEGGE 'MODO'
          STX MULTIC
          LDY #$00
          CPX #$02       ;SE>2, SALVA
          BCC CAR        ;SE=0 CARICA NORMALE
          LDY #$01       ;SE=1CARICA IN MCM
CAR       LDA #$02       ;FILE LOGICO 2
          LDX #$08       ;DEVICE 8
          JSR SETLFS     ;SETTA PARAMETRI FILE
          JSR $B7F1      ;LEGGI BYTE ALTO
          STX PAGE       ;PAGINA GRAFICA
          ;
          JSR CHKCOM     ;LEGGI NOME FILE
          JSR FRMEVL
          JSR LEN1
          TYA
          LDX $22        ;SETTA IL FILE NAME
          LDY $23
          JSR SETNAM
          ;
          JSR $B7F1      ;LEGGI ASCISSA
          CPX #40        ;CHE DEVE ESSERE
          BCC OKAY1      ; <40
;-------------------------------------
ERRORE    LDX #$0E       ;ILLEGAL QUANTITY
          JMP ($0300)
;-------------------------------------
OKAY1     STX XØ
          ;
          JSR $B7F1      ;LEGGI ORDINATA
          CPX #200       ;CHE DEVE ESSERE
          BCS ERRORE     ; <200
          STX YØ
          ;
          JSR $B7F1      ;LEGGI XL O FX
          STX XL
          STX FXØ
          STX FX
          ;
          JSR $B7F1      ;LEGGI YL O FY
          STX YL
          STX FYØ
          ;
          LDY SA         ;SE SECONDARY ADDRESS=0
          BEQ CARICAR    ;VAI A CARICARE
;-------------------------------------
          JSR OPEN       ;APRI UN FILE
          LDX #$02       ;IN SCRITTURA
          JSR CHKOUT
          ;
          LDA XL         ;SALVA DIMENSIONI
          JSR CHROUT     ;FINESTRA
          LDA YL
          JSR CHROUT
          ;
          LDY #$00       ;SALTA ALLA
          STY CONTFX     ;ROUTINE PRINCIPALE
          BEQ NXTLINE
;-------------------------------------
CARICAR   JSR OPEN       ;APRI UN FILE IN
          LDX #$02       ;LETTURA
          JSR CHKIN
          JSR CHRIN      ;LEGGI DIMENSIONI
```

```
         STA XL          ;FINESTRA
         JSR CHRIN
         STA YL
         ;
         LDX #$00        ;CONTROLLA PRESENZA
         JSR CHRGOT      ;PARAMETRO MASCHERA
         CMP #$2C        ;(DEFAULT=0)
         BNE NOMASK
         JSR $B7F1
NOMASK   STX MASK
         ;
         JSR CHRGOT      ;CONTROLLA PRESENZA
         CMP #$2C        ;PARAMETRI
         BNE NXTLINE     ;DIMENSIONI FINESTRA
         JSR $B7F1
         STX XL          ;LARGHEZZA
         ;
         JSR CHRGOT      ;E ALTEZZA
         CMP #$2C        ;(DEFAULT:QUELLO CHE
         BNE NXTLINE     ;HAI TROVATO SUL
         JSR $B7F1       ;DISCHETTO)
         STX YL
         ;
;-----------------------------------------
         ;
NXTLINE  LDA XL          ;CONTATORE PER
         STA CONTXL      ;LARGHEZZA FINESTRA
         LDA X0          ;ASCISSA INIZIALE
         STA X
         ;
NXTBYTE  LDA $90         ;ERRORI O FINE FILE?
         BEQ VAI
;-----------------------------------------
END      JSR CLRCHN      ;PULISCI CANALI
         LDA #$02        ;CHIUDI
         JMP CLOSE       ;E RITORNA
;-----------------------------------------
VAI      LDA SA          ;STIAMO SALVANDO?
         AND #%10011111  ;VAI A 'CALCOLA'
         BNE CALCOLA
         ;
         JSR CHRIN       ;LEGGI DA DISCO UN BYTE
         EOR MASK        ;EOR CON MASCHERA
         STA BYTE        ;'SALVALO'
         LDY FX0
         STY CONTFX
NEXTB    LDA #%00000001  ;INGRANDISCINE UNA
         STA BY          ;PARTE
         LDX FX
NEXTA    LDA BYTE
         LDY MULTIC      ;(A GRUPPI DI 2 BITS SE
MCM      ASL A           ;SIAMO IN MCM)
         ROL BY
         DEY
         BPL MCM
         DEX
         BNE OK1
         LDX FX0
         STA BYTE
OK1      BCC NEXTA
         STX FX
         LDX FY0
         DEC CONTFX
         CLC             ;VAI A CALCOLARE IN
         BCC CALCOLA     ;QUALE LOCAZIONE VA
                         ;POSTO IL BYTE CALCOLATO
CARICA   LDA BY          ;STAMPA IL BYTE
         STA (ADRS),Y
         DEX             ;CONTROLLA SE E' FINITO
         BNE COMPUT      ;L'INGRAND. VERTICALE
         ;
SALVA    LDA Y
OK2      INC X
         LDY CONTFX      ;CONTROLLO PER FINE
         BNE NEXTB       ;INGR.ORIZZ. DI 1 BYTE
         DEC CONTXL      ;CONTROLLO PER FINE
         BNE NXTBYTE     ;DI UNA RIGA DI BYTES
         STA Y0
         DEC YL          ;CONTROLLO FINE FINESTRA
         BNE NXTLINE     ;ANCHE IN VERTICALE
         BEQ END
;-----------------------------------------
CALCOLA  LDA Y0
         STA Y
COMPUT   LDA Y           ;SE L'ORDINATA E'
         CMP #200        ;>199, SOTTRAI 200
         BCC OK3
         SBC #200
         STA Y
OK3      AND #%11111000  ;Y AND 248
         STA ADRS
         LSR A           ;Y*32+PAGE
         LSR A
```

```
         LSR A
         ADC PAGE
         STA ADRS+1
         LDA X              ;SE L'ASCISSA E'
         CMP #40            ;>39, SOTTRAI 40
         BCC OK4
         SBC #40
         STA X
         CLC
OK4      ADC ADRS           ;(X+(YAND248))*8+
         STA ADRS           ;   +(YAND248)*32+PAGE
         LDA #$00
         ASL ADRS
         ROL A
         ASL ADRS
         ROL A
         ASL ADRS
         ROL A
         ADC ADRS+1
         STA ADRS+1
         LDA Y              ;Y AND 7
         AND #%00000111
         TAY
```

```
         INC Y
;----------------------------------------
         LDA SA             ;STIAMO CARICANDO?
         AND #%10011111     ;VAI A 'CARICA'
         BEQ CARICA
         ;
         LDA $DC0E          ;SPEGNI INTERRUPT
         TAX
         AND #%11111110
         STA $DC0E
         LDA $01
         AND #%11111000     ;SCOPRI TUTTA LA RAM
         STA $01
         LDA (ADRS),Y       ;LEGGI IL VALORE
         TAY
         LDA $01
         ORA #%00000111     ;COPRI LA RAM
         STA $01
         STX $DC0E          ;RIABILITA INTERRUPT
         TYA
         JSR CHROUT         ;SALVA IL VALORE
         CLC                ;TORNA ALLA ROUTINE
         BCC SALVA          ;PRINCIPALE
```

I componenti "SPEEDDOS" vengono scaricati a bordo; dopo aver trovato una presa di corrente sufficiente anche per il saldatorone, l'equipaggio inizia il lavoro.
Primo Giovedini, salito in plancia, coordina le operazioni...

Ready, ready...

Hei, voi due! Mettete un po' più di stagno!

Sys, hai sentito il capo? Muoviamoci...

BZZZZZZZ!

LA CRESCITA DEI CARATTERI

Una routine in l.m. consente di ingrandire a dismisura qualsiasi carattere alfa-numerico-grafico; anche se personalizzato

di **Tito Brasolin**

Spesso può essere utile inserire, negli sprite, uno o più caratteri alfanumerici, o grafici, per visualizzare messaggi ingranditi e in movimento. Questo è ciò che può fare il programma proposto che controlla l'ingrandimento dei caratteri e la loro posizione in un qualsiasi sprite.

LA SINTASSI DA USARE

Il formato della nuova istruzione è il seguente:

Sys xxxxx, cc, sn, x, y, fx, fy [,bank[, memcar]

...in cui:

- CC = è il Codice del Carattere da inserire nello sprite. Varia da zero a 65535.
- SN = Numero dello Sprite, compreso fra zero e 255.
- X = Ascissa dell'angolo superiore sinistro del carattere. Varia da zero a 23.
- Y = Ordinata dell'angolo superiore sinistro del carattere. Varia da zero a 21.
- FX = Fattore di ingrandimento orizzontale.
- FY = Fattore di ingrandimento verticale. Sia questo parametro che il precedente sono compresi tra zero e 255.
- BANK = Banco da 16K cui si riferisce il numero dello sprite. E' facoltativo, può variare tra zero e tre ed è uguale a zero per default.
- MEMCAR = Indirizzo locazione iniziale del generatore dei caratteri. Facoltativo, compreso tra zero e 65535, risulta uguale a 53248 per default.

IL CODICE CARATTERE

Il C/64 dispone di due set di 256 caratteri ciascuno, per un totale di 512. Si sceglie quale di questi inserire nello sprite tramite il parametro CC, i cui valori utili sono dunque solo quelli fino a 511 (zero compreso). Valori maggiori, pur non generando errore, rappresenteranno una "ripetizione" dei set.

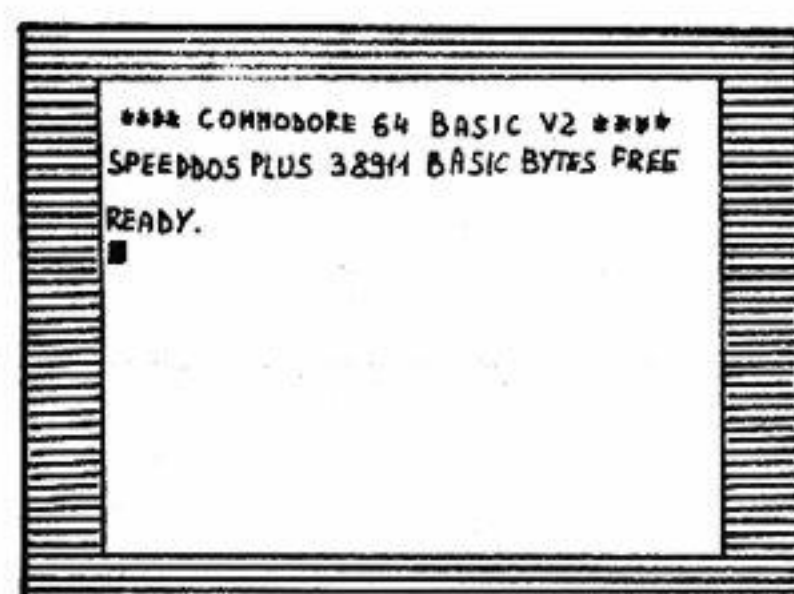
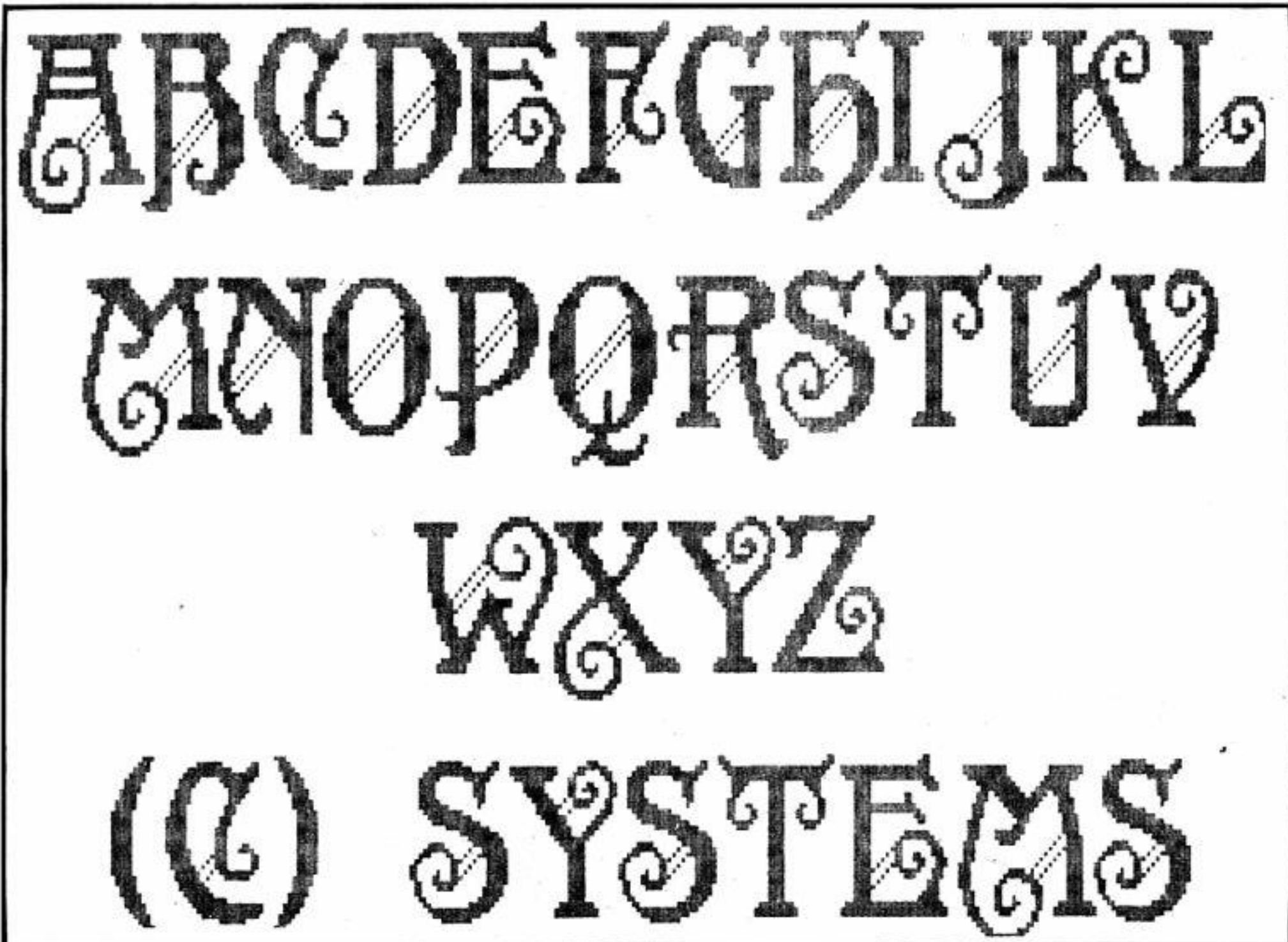
Per calcolare il codice da fornire alla routine, basatevi sulla tabella dei CODICI SCHERMO riportata nel manuale del computer. Questa fornisce un numero compreso tra 0 e 127, al quale dovrete sommare 128 (per ottenere un carattere in reverse) oppure 256 (se deve appartenere al set minuscolo / maiuscolo). Per esempio il codice di una "T" minuscola in reverse è pari a 20 + 128 + 256 = 404.

La routine, in seguito, moltiplica tale valore per 8 e somma il risultato alla locazione iniziale del generatore dei caratteri (normalmente posizionato a partire da 53248), ottenendo così l'indirizzo a partire dal quale sono fisicamente presenti gli 8 bytes del carattere in questione.

IL NUMERO DI SPRITE

Tramite il parametro SN si decide in quale sprite debba esser trascritto il carattere. Esso varia tra zero e 255, e siccome uno sprite occupa 64 bytes è possibile spaziare in 16 k di memoria:

64 * 256 = 16384

ABCDEFGHIJKL
MNOPQRSTUV
WXYZ
(C) SYSTEMS

La locazione di inizio dello sprite è data dalla formula:

SN * 64 + BANK * 16384

Dal momento che la routine è INTERAMENE rilocabile, è indispensabile prestare la massima attenzione alla zona Ram in cui verranno trascritti gli sprite; potrebbe capitare, infatti, che questi "cadano" sulla pagina 0, sull'area del programma Basic o sulla stessa routine l.m. qui descritta. In questi casi si rischia l'inchiodamento della macchina.

IL BANCO

Con il parametro BANK, normalmente posto a zero, si decide quale blocco da 16K considerare:

BANK = 0: 0 / 16384
BANK = 1: 16385 / 32767
BANK = 2: 32768 / 49151
BANK = 3: 49152 / 65535

LE COORDINATE

Uno sprite è largo 24 bits e alto 21. E' quindi logico che il valore delle coordinate dell'angolo superiore sinistro del carattere possa variare da (0, 0) (corrispondente all'angolo superiore sinistro dello sprite) a (23, 20) (corrispondente all'angolo inferiore destro). Superando determinati valori il carattere uscirà, in parte, dallo sprite.

L'aver garantito tale precisione nel posizionamento dello sprite richiede alcuni calcoli, simili a quelli che si effettuano per il plottaggio di punti in una pagina grafica. Poniamo infatti che sia SPRITE

Potete far apparire caratteri maiuscoli, minuscoli e grafici, anche in reverse

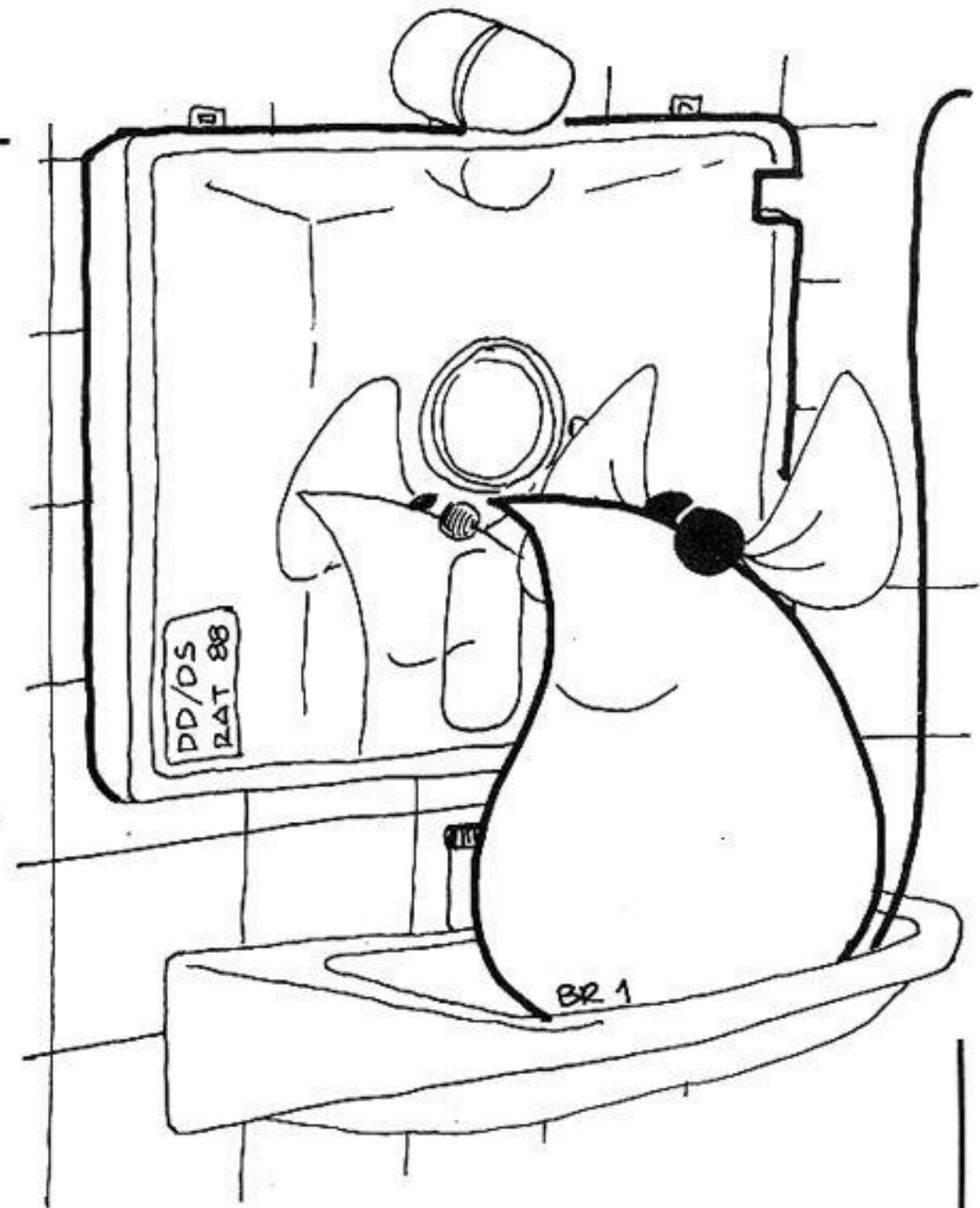
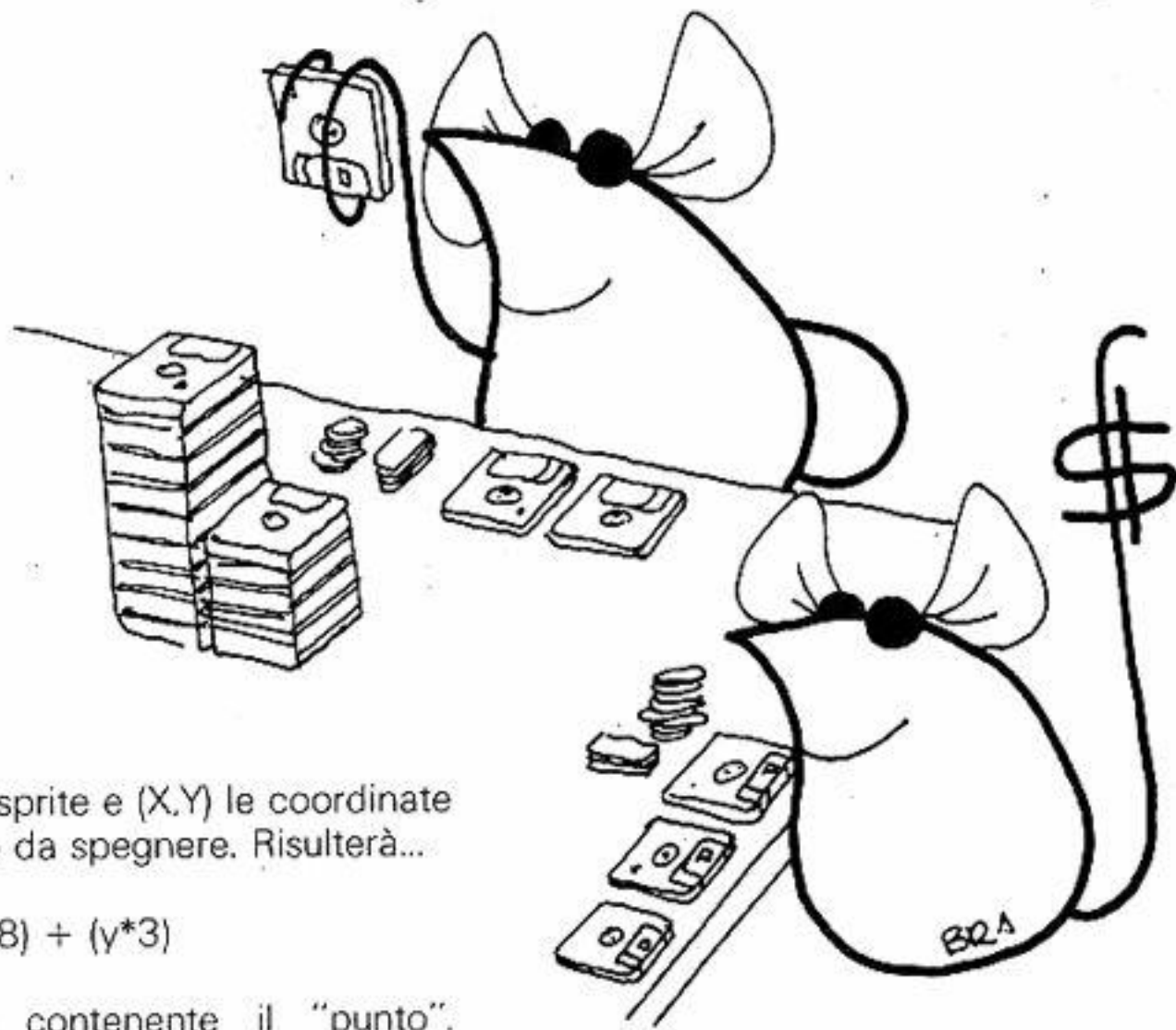
Primo Giovedini è pronto al timone dell'incrociatore AEGIS ...
Attenzione, ora partiamo!
F3!
TURBO!
Restore! Stiamo già andando a 50 MHz!
VROOOOOOOOMM

```
;-----------------------------------
;
;       CARATTERI NEGLI SPRITE
;
;         BY TITO BRASOLIN
;
;            GIUGNO 1989
;
;     DISASSEMBLATO COMMENTATO
;-----------------------------------
CHKCOM  = $AEFD
FRMNUM  = $AD8A
GETADR  = $B7F7
GETNUM  = $B7EB
CHRGOT  = $79
        ;
INDEX   = $02
BYTNUM  = $F7
MASK    = $F8
X       = $F9
Y       = $FA
SPRITE  = $FB
CHAR    = $FD
Y0      = $0313
FX      = $0334
FY      = $0335
TEST    = $0336
CONTFX  = $0337
;-----------------------------------
        JSR CHKCOM      ;SALTA UNA VIRGOLA
        JSR GETNUM      ;LEGGI CODICE CARATTERE
        STX SPRITE+1    ;E NUMERO SPRITE
        LDA $14
        STA CHAR
        LDA $15
        STA CHAR+1
        ;
        JSR $B7F1       ;LEGGI ASCISSA
OKX     STX X
        ;
        JSR $B7F1       ;LEGGI ORDINATA
        STX Y0
        ;
        JSR $B7F1       ;LEGGI FATTORE
        STX FX          ;INGR.ORIZZ.
        ;
        JSR $B7F1       ;LEGGI FATTORE
        STX FY          ;INGR.VERT.
        ;
        LDX #$00        ;0=DEFAULT PER
        JSR CHRGOT      ;IL BANCO
        CMP #$2C        ;SE C'E' UNA
        BNE BENE        ;ALTRA VIRGOLA,
                        ;LEGGI IL NUOVO
        JSR $B7F1       ;VALORE DEL BANCO
        CPX #4          ;SE E' >3
        BCC BENE
        LDX #$0E        ;ILLEGAL QUANTITY
        JMP ($0300)
        ;
BENE    TXA             ;CALCOLA
        LSR A           ;#SPRITE*64+BANK
        ROR SPRITE+1
        ROR A
        ROR SPRITE+1
        STA SPRITE
        ;
        LDX #$00        ;$D000=DEFAULT
        STX $14         ;PER INIZIO SET
        LDX #$D0        ;CARATTERI
        STX $15
        JSR CHRGOT      ;SE C'E' UNA ALTRA
        CMP #$2C        ;VIRGOLA, LEGGI
        BNE MEMCAR      ;IL NUOVO INDIRIZZO
        ;               ;INIZIALE DEL SET
        JSR CHKCOM
        JSR FRMNUM
        JSR GETADR
        ;
MEMCAR  LDA CHAR        ;CALCOLA
        ASL A           ;CODICE*8+CARSET
        ROL CHAR+1
        ASL A
        ROL CHAR+1
        ASL A
        ROL CHAR+1
        CLC
        ADC $14
        STA CHAR
        LDA CHAR+1
        ADC $15
        STA CHAR+1
;-----------------------------------
        LDA $DC0E        ;DISABILITA
        AND #%11111110   ;INTERRUPT
        STA $DC0E
        LDA $01          ;SCOPRE GENERATORE
        AND #%11111001   ;CARATTERI E RAM
        STA $01          ;SOTTO IL S.O.
        ;
        LDA #%10000000   ;INIZIALIZZA MASCHERA
        STA TEST         ;PER TESTARE I BITS
                         ;DEL CARATTERE
                         ;
```

PUBBLICITA'..!

Lavorare al computer vi stanca gli occhi?

Non ne potete più di fosfori verdi ...

```
NEXT4   LDX FX              ;CONTATORE LARGHEZZA
        STX CONTFX          ;NUOVO 'PIXEL'
        ;
NEXT3   LDA X
        AND #%00000111
        TAY                 ;CALCOLA
        LDA #%10000000      ;2↑7-(X AND 7)
NEXT    DEY                 ;OTTENENDO
        BMI CONT            ;LA MASCHERA CON
        LSR A               ;CUI ACCENDERE
        BCC NEXT            ;IL BIT NELLO SPRITE
CONT    STA MASK
        ;
        LDA Y0              ;INIZIALIZZA ORDINATA
        LDY #$00            ;E PUNTATORE AL
NEXT2   STY BYTNUM          ;BYTE IN CORSO DEL
                            ;CARATTERE
                            ;
        LDX FY              ;CONTATORE PER INGR.VERT.
NEXT1   STA Y
        CMP #21             ;CONTROLLA CHE NON
        BCS CTRL            ;SIAMO FUORI DELLO
        STA INDEX           ;SPRITE
        LDA X               ;(IN BASSO O A DESTRA)
        CMP #24
        BCS EXIT
        LSR A               ;CALCOLA
        LSR A               ;INT(X/8)+Y*3
        LSR A
        ASL INDEX
        ADC Y
        ADC INDEX
        STA INDEX
        ;
        LDY BYTNUM
        LDA (CHAR),Y
        ;
        LDY INDEX
        AND TEST            ;TESTA IL BIT DEL
        BEQ CANCEL          ;CARATTERE
        LDA MASK            ;E ACCENDINE
        ORA (SPRITE),Y
        BNE STAMP
CANCEL  LDA MASK            ;O SPEGNINE UNO NELLO
        EOR #%11111111      ;SPRITE
        AND (SPRITE),Y
STAMP   STA (SPRITE),Y
        INC Y               ;AGGIORNA PUNTATORE AL
                            ;BYTE DEL CARATTERE
        LDY BYTNUM
        LDA Y
        DEX                 ;SE HAI FINITO
        BNE NEXT1           ;L'INGRANDIMENTO IN
        INY                 ;VERTICALE, PROCEDI
        CPY #$08            ;IN ORIZZONTALE
        BCC NEXT2
CTRL    INC X               ;SE NON HAI FINITO
        DEC CONIFX          ;IN ORIZZONTALE
        BNE NEXT3           ;AGGIORNA LA MASCHERA
        LSR TEST            ;PER TESTARE E PROSEGUI
        BCC NEXT4
                            ;ALTRIMENTI FINE
EXIT    LDA $01             ;COPRI GENERATORE
        ORA #%00000110      ;CARATTERI E RAM SOTTO
        STA $01             ;IL S.O.
        LDA $DC0E           ;RIABILITA
        ORA #%00000001      ;INTERRUZIONI
        STA $DC0E
        ;
        RTS                 ;RITORNA
```

l'indirizzo iniziale dello sprite e (X,Y) le coordinate del punto da plottare o da spegnere. Risulterà...

Adrs = sprite + int (x/8) + (y*3)

...l'indirizzo del byte contenente il "punto". Inoltre, se...

Mask = 2 exp (7-(x and 7))

...si imporrà...

Poke adrs, peek (adrs) or mask

...per plottare il punto, e...

Poke adrs, peek (adrs) and (255-mask)

...per spegnerlo.

L'INGRANDIMENTO E "MEMCAR"

I parametri FX e FY definiscono il numero di volte per cui deve essere ingrandito il carattere inserito nello sprite.

Ciascun bit del carattere viene trasformato dalla routine in un rettangolo del formato di FX x FY bits.

Anche se possono variare entrambi tra 0 e 255 (con lo zero che ha l'effetto del valore 256), solo pochi valori generano un carattere che può essere contenuto interamente nello sprite. Il massimo in-

ESPANSIONI

I lettori più in gamba possono sfruttare altre tecniche, tra cui la programmazione di 64 sprite contemporanei, per visualizzare messaggi fortemente suggestivi.

Chi se la sente può tentare di utilizzare, oltre a quella proposta in queste stesse pagine, altre routine già pubblicate, come ad esempio quella presente sul fascicolo N. 63 (pagina XXI "Come gestire 64 sprite").

La routine ideale è quella che, utilizzando le routine già pubblicate (SENZA apportarvi alcuna modifica!) sia in grado di fonderle tra loro e di sfruttarne le peculiarità grazie ad un sistema semplice ed efficace.

Se pervenite a qualcosa di interessante, telefonateci (02 / 52.49.211) per determinare le modalità di pubblicazione del vostro lavoro.

grandimento "intero" orizzontale è 3, giacchè uno sprite è largo 24 bytes ed un carattere 8; infatti: 8 x 3 = 24.

Con FY=3 una parte del carattere "esce", essendo uno sprite alto solo 21 bits. Talvolta questa limitazione non genera problemi; poichè la maggior parte dei caratteri è contenuta nei primi sette bytes mentre l'ottavo è lasciato vuoto per aumentare la leggibilità su schermo.

Ma in altri casi si perdono, ad esempio, i discendenti delle lettere minuscole. Si può ovviare a tale inconveniente agendo sul parametro MEMCAR. Se lo si pone uguale a 53249 la routine leggerà, di ogni carattere, solo gli ultimi 7 bytes, oltre al primo del carattere successivo (che non interessa). I caratteri saranno shiftati in alto e i discendenti rientreranno nello sprite, anche nel caso in cui vengano moltiplicati per tre in verticale. Stavolta, però, usciranno le "cime" delle lettere maiuscole, per cui bisognerà regolarsi caso per caso.

L'uso principale del parametro MEMCAR serve per ordinare alla routine di fare riferimento ad un set di caratteri definiti dall'utente: è sufficiente impostare MEMCAR con l'indirizzo iniziale di tale set.

I dati possono trovarsi in qualsiasi zona RAM, compresa quella posizionata sotto il Kernal, ma NON in quella sotto l'interprete Basic. Questo perchè la routine provvede a "scoprire" la parte alta della RAM e il generatore dei caratteri (poke 1, peek (1) and 249), ma se si tentasse di evidenziare anche la RAM sotto il Basic (poke 1, peek (1) and 248) sparirebbe anche il generatore di caratteri, e verrebbero resi disponibili i 64K di RAM, il cui contenuto potrebbe riservare amare sorprese.

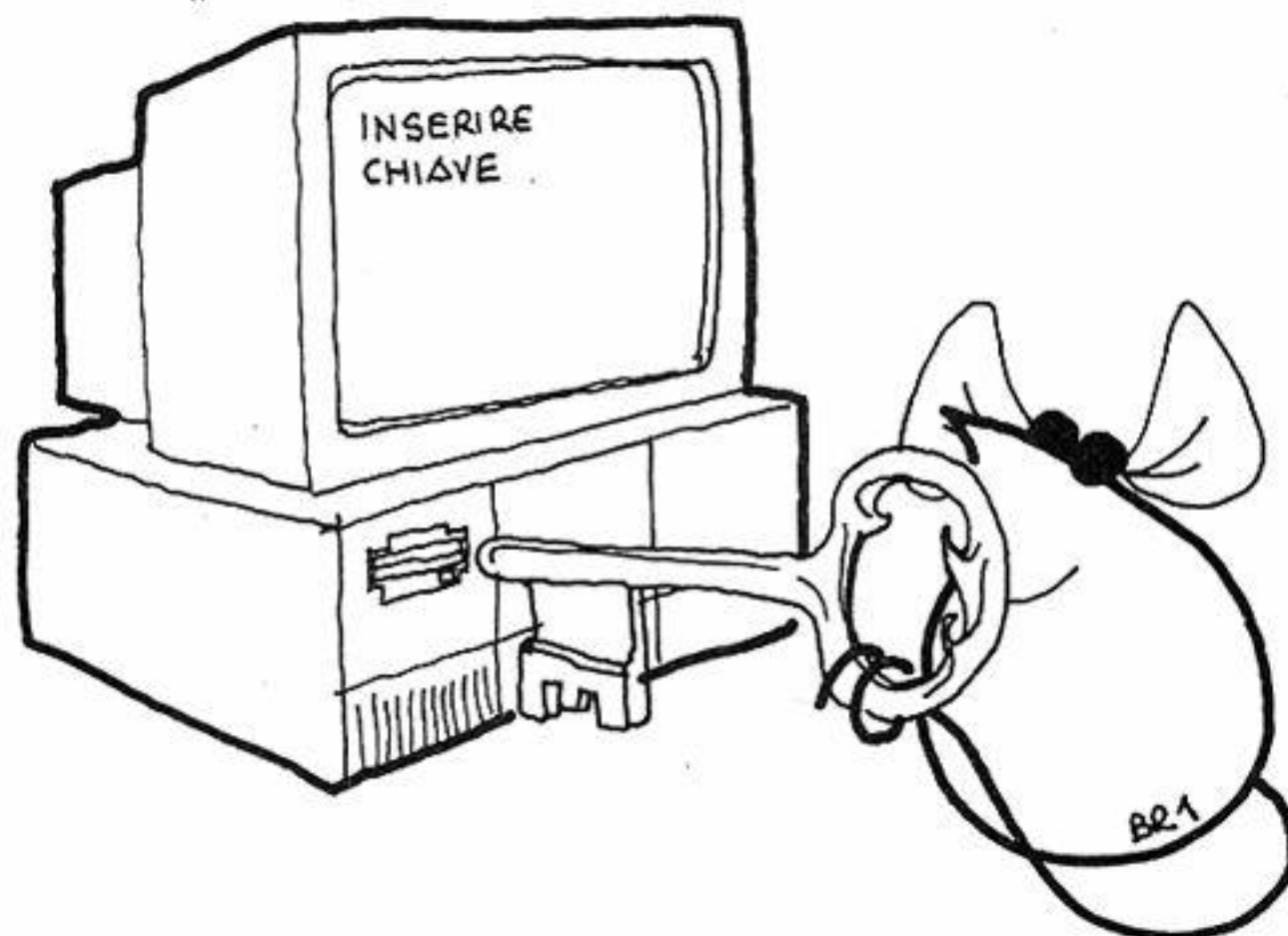
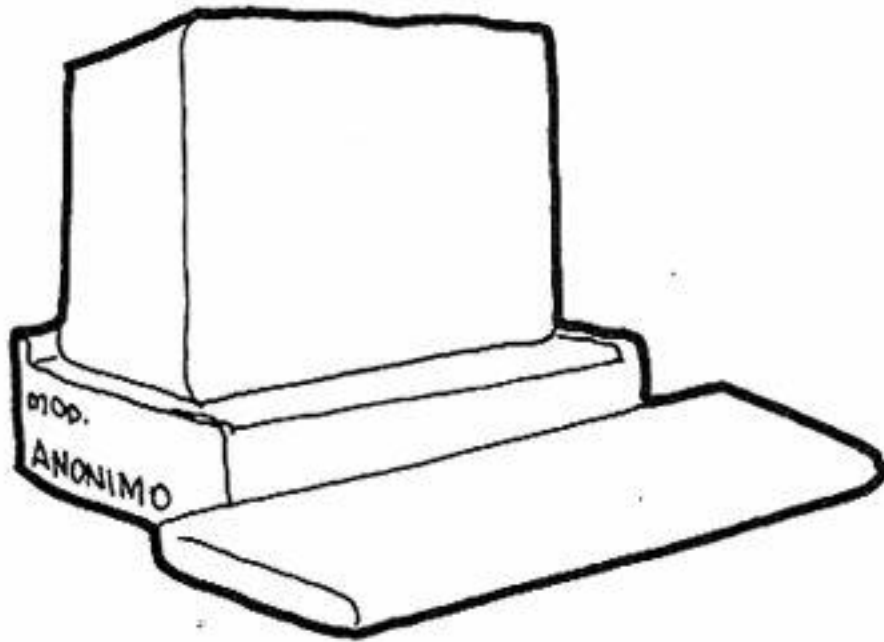
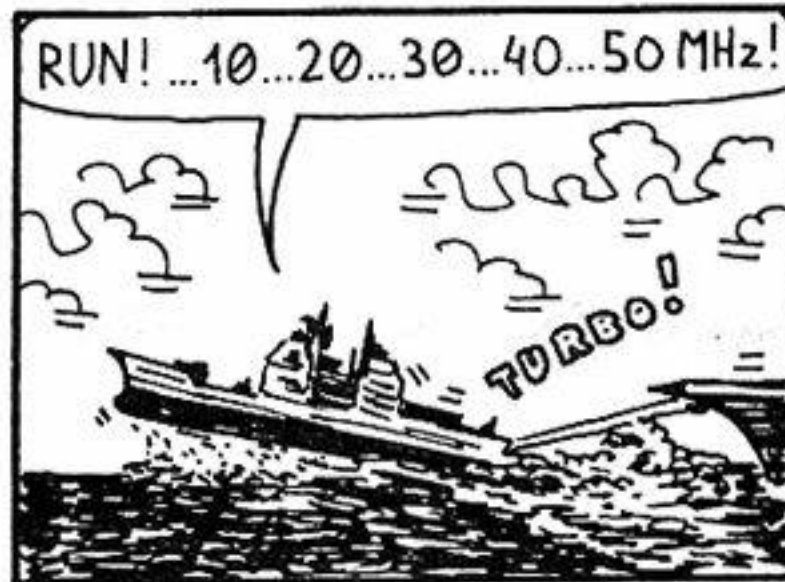
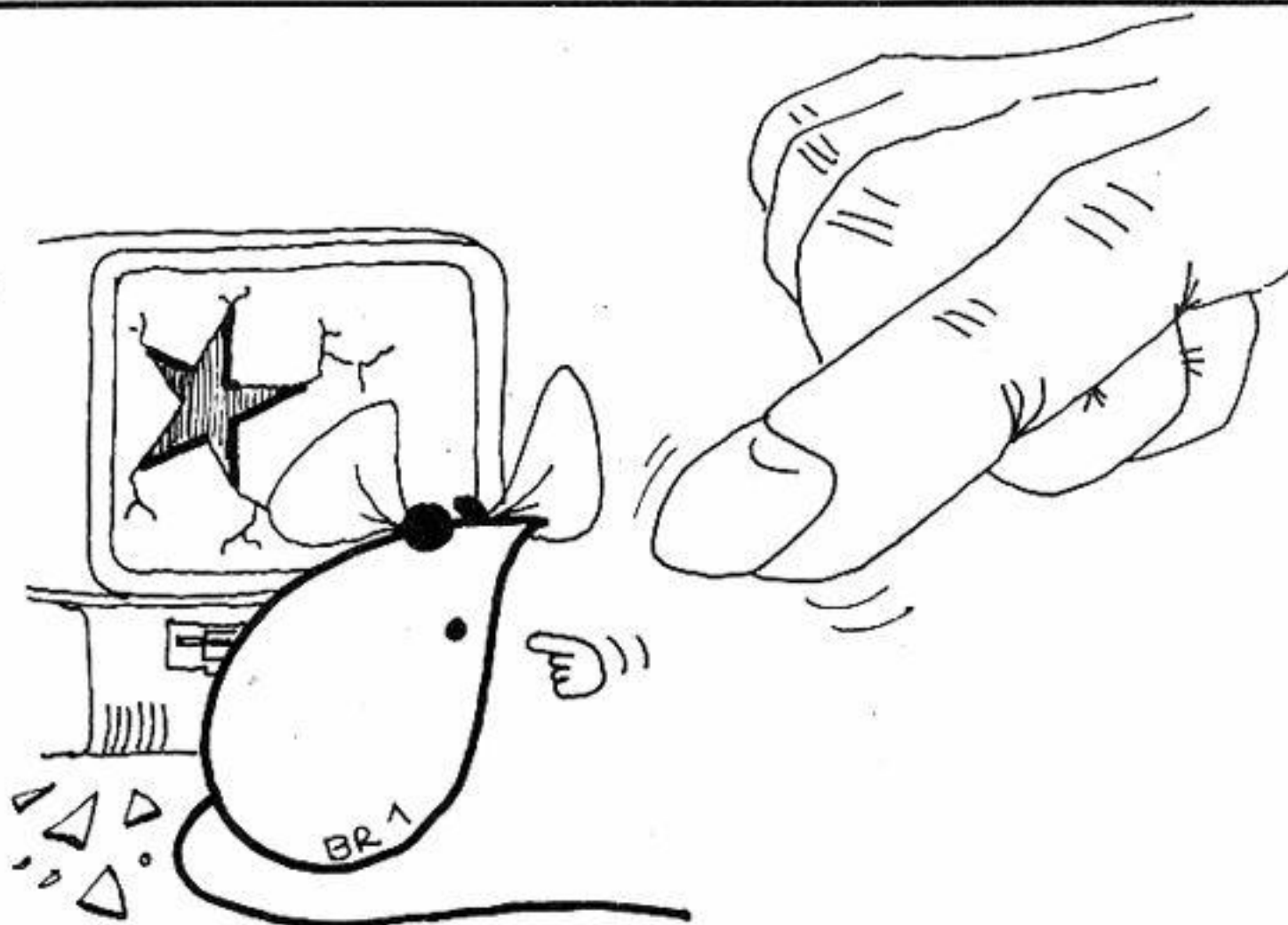
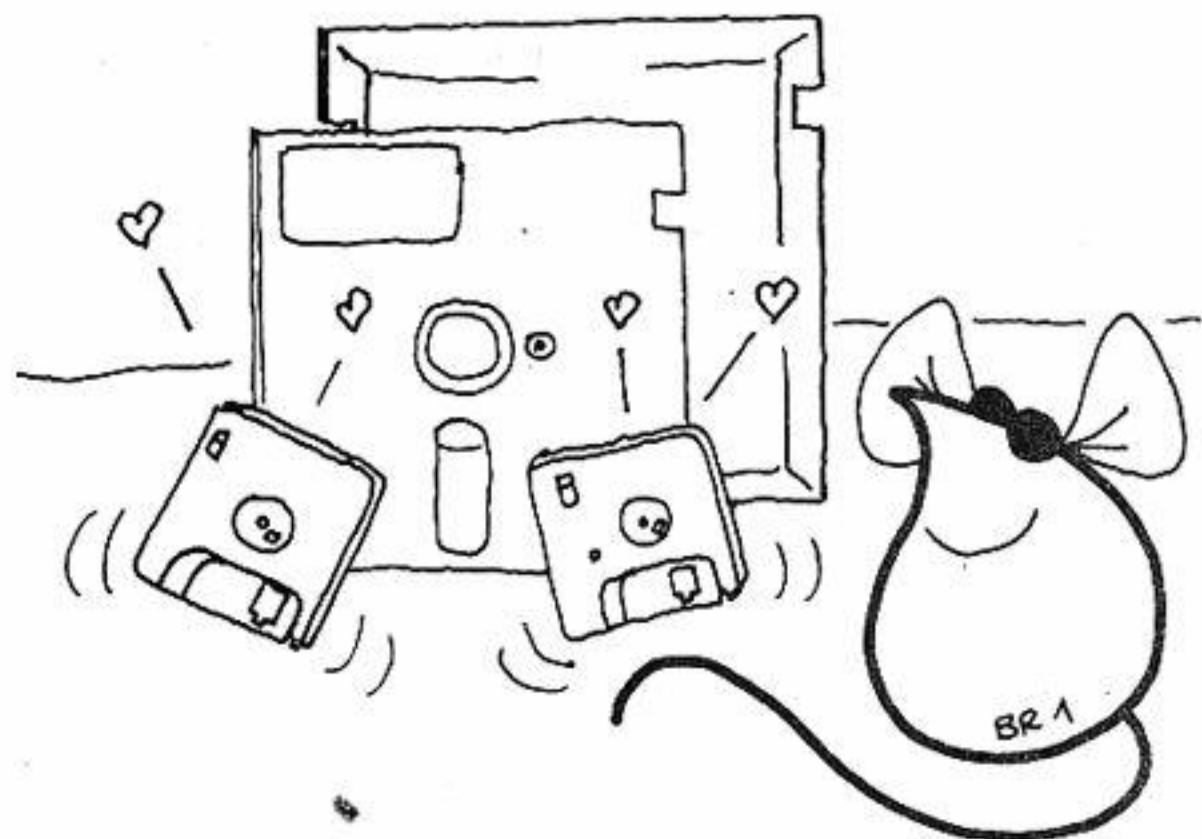
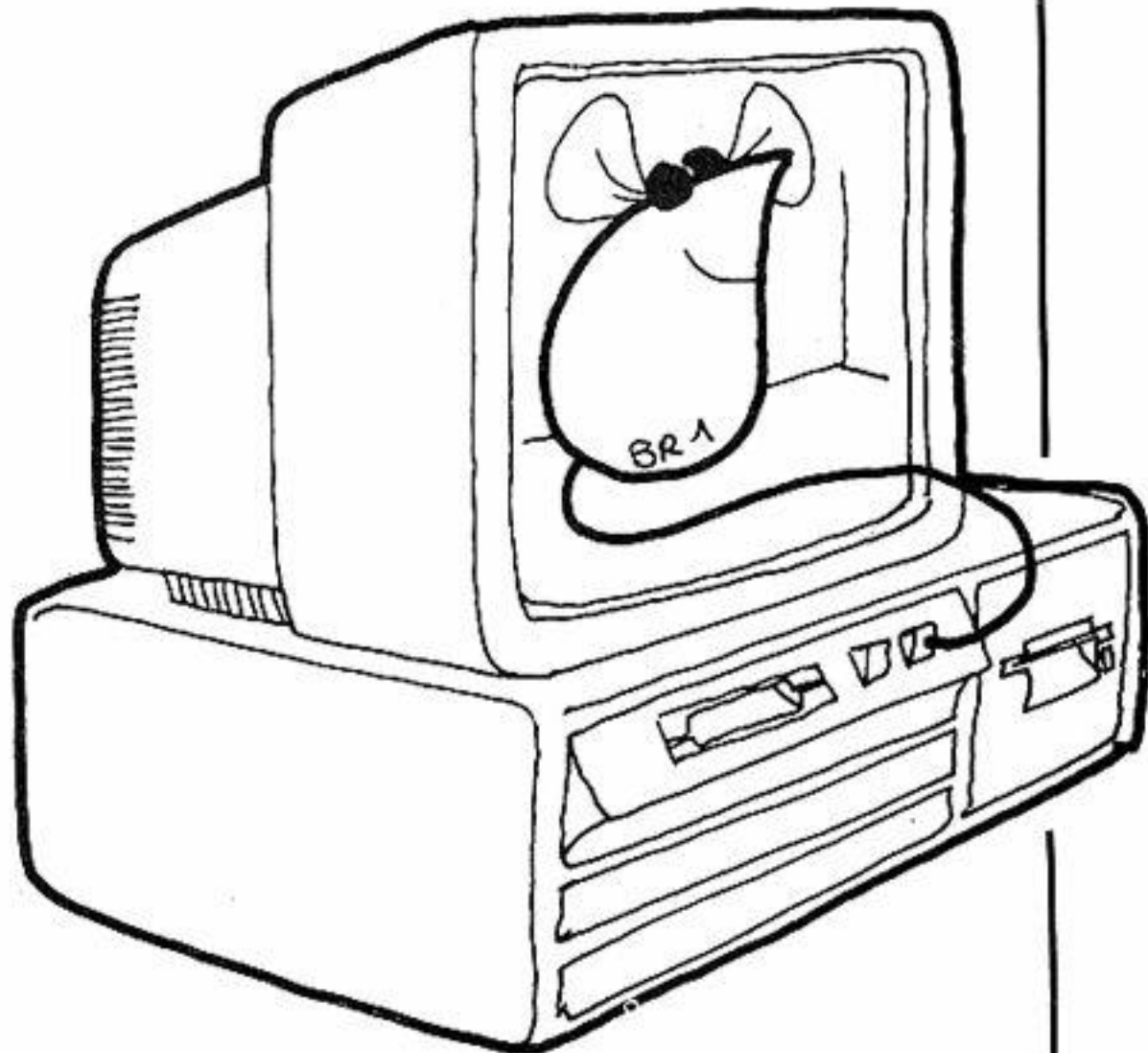
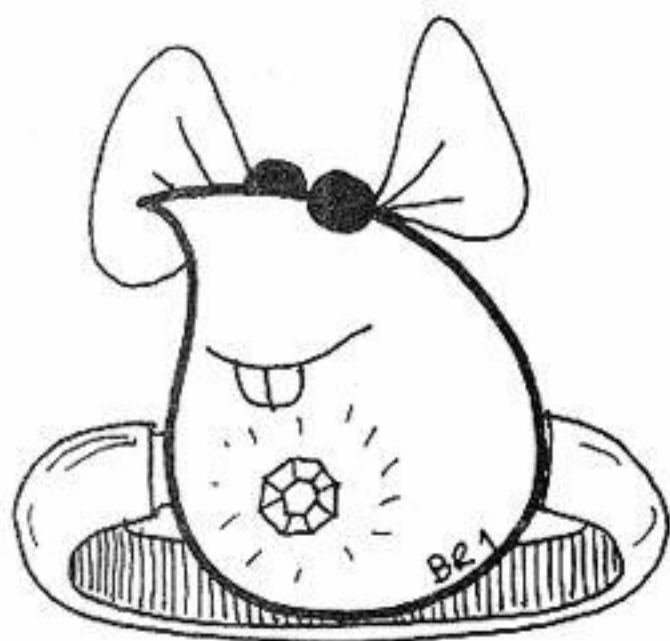
I DEMO

I due listati dimostrativi pubblicati richiedono entrambi che la routine sia presente in memoria, allocata a partire da 49152.

Il primo programma mette in evidenza la precisione possibile nel posizionamento di un carattere in uno sprite. E' il semplice effetto della pallina rimbalzante, ottenuto spostando un carattere (pallina in negativo) all'interno di uno sprite. La pallina, muovendosi, scopre anche il disegno presente nello sprite sottostante.

Il secondo demo, mediante un input iniziale, richiede una stringa (lunga al massimo 8 caratteri, tanti quanti sono gli sprite) i cui caratteri vengono, appunto, inseriti negli sprite cambiandone varie volte le dimensioni.

Ricordiamo ai più distratti che gli 8 sprite vengono visualizzati affiancati tra loro. Questo è un caso particolare che potrebbe inchiodare il sistema nel caso in cui, con gli 8 sprite presenti su video, si tentassero operazioni di Input / Output con il drive. Prima di caricare (o registrare) file, pertanto, "spegnete" gli sprite, magari con la semplice pressione dei tasti Run / Stop e Restore.

Anche i principianti possono usare la routine: digitatela con la massima attenzione

```
100 REM LA CRESCITA DEI CARATTERI.
110 REM ROUTINE L.M. INTERAMENTE RILOCABILE
120 REM PER COMMODORE 64
130 REM LINGUAGGIO MACCHINA
150 REM PER UTILIZZARLA, LEGGERE LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
180 Y=-1:GOSUB 1000:PRINT:PRINT"(DEFAULT = 49152)":X= 49152
185 INPUT"DA QUALE LOCAZIONE";X
190 READW:Y=Y+1:IFW<0THEN210
200 GOTO190
210 PRINT"PRIMA LOCAZIONE ="X
220 PRINT"ULTIMA LOCAZIONE ="X+Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GETA$:IFA$=""THEN240
250 IFA$="S"THENRESTORE:T=X:GOTO280
260 RUN
270 :
280 GOSUB1000:PRINT:PRINT"ATTENDERE...":PRINT:W=0
290 READB:IFB>=0THENPOKET,B:T=T+1:W=W+B:GOTO290
300 READB:IFB<>WTHENPRINT:PRINTCHR$(18)"ERRORE DI TRASCRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X"A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PRINT
330 X1=INT(X/256):X2=X-(X1*256):PRINT"POKE44,"X1":POKE43,"X2;
340 Y1=INT(T/256):Y2=T-(Y1*256):PRINT":POKE46,"Y1":POKE45,"Y2:END
1000 PRINT CHR$(147);
1010 PRINT "CARATTERI NEGLI SPRITE":PRINT
1020 PRINT "USO: SYS XXXX,CC,SN,X,Y,FX,FY [,BANK     [,MEMCAR]]":PRINT
1030 PRINT "XXXX  =INDIRIZZO DI PARTENZA QUALSIASI"
1040 PRINT "CC    =CODICE DEL CARATTERE (0...65535)"
1050 PRINT "SN    =SPRITE NUMBER (0...255)"
1060 PRINT "X     =ASCISSA DEL CARATTERE (0...23)"
1070 PRINT "Y     =ORDINATA DEL CARATTERE (0...20)"
1080 PRINT "FX    =FATTORE INGR. ORIZZ. (0..255)"
1090 PRINT "FY    =FATTORE INGR. VERT. (0..255)"
1100 PRINT "BANK  =BANCO(0..3).OPZIONALE: DEFAULT=0"
1110 PRINT "MEMCAR=INIZIO SET CARATTERI (0...65535)"
1120 PRINT "       OPZIONALE: DEFAULT=53248"
1130 RETURN
```

```
1140 DATA 032,253,174,032,235,183,134
1150 DATA 252,165,020,133,253,165,021
1160 DATA 133,254,032,241,183,134,249
1170 DATA 032,241,183,142,019,003,032
1180 DATA 241,183,142,052,003,032,241
1190 DATA 183,142,053,003,162,000,032
1200 DATA 121,000,201,044,208,012,032
1210 DATA 241,183,224,004,144,005,162
1220 DATA 014,108,000,003,138,074,102
1230 DATA 252,106,102,252,106,133,251
1240 DATA 162,000,134,020,162,208,134
1250 DATA 021,032,121,000,201,044,208
1260 DATA 009,032,253,174,032,138,173
1270 DATA 032,247,183,165,253,010,038
1280 DATA 254,010,038,254,010,038,254
1290 DATA 024,101,020,133,253,165,254
1300 DATA 101,021,133,254,173,014,220
1310 DATA 041,254,141,014,220,165,001
1320 DATA 041,249,133,001,169,128,141
1330 DATA 054,003,174,052,003,142,055
1340 DATA 003,165,249,041,007,168,169
1350 DATA 128,136,048,003,074,144,250
1360 DATA 133,248,173,019,003,160,000
1370 DATA 132,247,174,053,003,133,250
1380 DATA 201,021,176,058,133,002,165
1390 DATA 249,201,024,176,062,074,074
1400 DATA 074,006,002,101,250,101,002
1410 DATA 133,002,164,247,177,253,164
1420 DATA 002,045,054,003,240,006,165
1430 DATA 248,017,251,208,006,165,248
1440 DATA 073,255,049,251,145,251,230
1450 DATA 250,164,247,165,250,202,208
1460 DATA 197,200,192,008,144,187,230
1470 DATA 249,206,055,003,208,160,078
1480 DATA 054,003,144,149,165,001,009
1490 DATA 006,133,001,173,014,220,009
1500 DATA 001,141,014,220,096
1510 DATA -1,31385
1520.END
```

```
10 REM * CARATTERI NEGLI SPRITE
20 REM * BY TITO BRASOLIN
30 REM * GIUGNO 1989
40 REM * DIMOSTRAZIONE N.1
50 REM * (PER ROUTINE ALLOCATA DA 49152)
60 :
70 PRINT CHR$(147);"       PALLINA RIMBALZANTE"
80 V=53248
90 POKE V,24:POKE V+1,50
100 POKE V+2,24:POKE V+3,50
110 POKE V+23,3:POKEV+29,3
120 POKE V+39,4:POKEV+40,7
130 POKE V+21,7
140 POKE 2040,13:POKE 2041,14
150 REM FOR X=0 TO 16 STEP 8:FOR Y=0 TO 16 STEP 8
160 SYS 49152,1,14,0,0,3,3
170 REM NEXT Y:NEXT X
180 SYS 49152,32+128,13,0,0,3,3
190 X=0:Y=0
200 DX=1:DY=1
210 IF X<0 OR X>16 THEN DX=-DX:X=X+DX
220 IF Y<0 OR Y>12 THEN DY=-DY:Y=Y+DY
230 SYS 49152,81+128,13,X,Y,2,2
240 X=X+DX:Y=Y+DY:GOTO210
250 END
```

```
10 REM * CARATTERI NEGLI SPRITE
20 REM * BY TITO BRASOLIN
30 REM * GIUGNO 1989
40 REM * DIMOSTRAZIONE N.2
50 REM * (PER ROUTINE ALLOCATA DA 49152)
60 POKE 55,0:POKE 56,32
70 PRINT CHR$(147);:INPUT "SCRIVI QUALCOSA";A$:PRINTCHR$(147);A$
80 V=53248
90 REM * POSIZIONA E PULISCE GLI SPRITES
95 REM R DEVE ESSER MINORE DI 8!!!
100 FOR R=0 TO 7: SYS 49152,32,128+R,0,0,3,3
110 POKE V+2*R,24*(R+1)
120 POKE V+2*R+1,98
130 POKE V+39+R,3
140 POKE 2040+R,128+R
150 NEXT R: POKE V+21,255
160 REM * FASE DI INGRANDIMENTO
170 FOR FX=1 TO 3:REM PROVARE CON... TO 5
180 FOR FY=1 TO 3:REM PROVARE CON... TO 5
190 FOR R=0 TO 7:REM PROVARE CON... TO 10
200 SYS 49152,PEEK(1024+R),128+R,0,0,FX,FY
210 NEXT R: REM FOR HH=1 TO 1000: NEXT
220 NEXT FY:REM EVENTUALE RITARDO
230 NEXT FX
240 RUN
```

Finalmente, il salvataggio è stato eseguito.
A modo suo, Primo Giovedini è riuscito a rimediare al BUG che rischiava di inchiodare per sempre la sua portaerei!
Arrivederci al prossimo FILE.

SYS 64738

Vuoi pubblicare un annuncio gratuito?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:
Commodore Computer Club - Viale Famagosta, 75 - 20142 Milano

Ultime novità Amiga con il solo costo del disco!! Si solo c/o il S.A.C.
Telefonare allo 0362/501857 dalle ore 20.00 alle ore 21.00, chiedere di Luca. A presto!!!

Cerco stampante MPS 801/802 o 803 per C64 usata (funzionante) in zona Varese - Novara.
Piergiorgio Lepori - Via Torino 4/A - 28060 Cureggio (NO) - Telefonare dopo le ore 20.00 allo 0332/809679

Vendo in ottime condizioni (quasi nuovi) drive 1571 per C64/C128 200.000,stampante Okimate 20 color 275.000.
Scrivere a Doddis Roberto - V.le R. Elena is. 107 Pal.D'Int. 92 - 98100 (ME)

Cerco per C64: drive - stampante e riviste varie (con cassetta) arretrate come: SuperCommodore 64; radio elettronica & C; Com.Comp.Cl. a prezzi modici da concordare.
Scrivere a Di Giovanni - Via Tremiti n. 1 - 86046 San Martino (CB)

Vendo software di ogni tipo per C64, a prezzi irrisori, solo 3.000 a disco, in tutt'Italia.
Max serietà, sconti per ordini sopra i 15 dischi.
Contattarmi non costa nulla, aspetto notizie!
2 dischi in regalo per un primo ordine, tutte le novità.
Simone Galigani - Via F.D. Guerrazzi n. 8/A - S.P. in Palazzi (LI) - Tel. 0586/660070

Vendo 2 penne ottiche + 2 registratori + 11 dischetti per Drive 1541 + duplicatore per cassette e 40 stupende cassette di giochi vari, tutto per C64 a 250.000 + un regalo a sorpresa.
Tel. 9267814 (ore pasti)

Vendo ATARI 2600 (completo di trasformatore + cavo antenna) + 20 giochi circa a 70.000 (è un'affare!) causa passaggio Amiga.
Telefonare allo 059/692861 nelle ore pomeridiane e chiedere di Andrea.

Vendo cassette piene di videogames & Utility per C64 a 1.000 - 2.000 l'una!!
Fornara Giorgio - Via Cialdini n. 6 - 20022 Castano Primo (MI) - Tel. 0331/881323

Attenzione: per C64 cerco manuali di Ultima V scambio con Rocket Ranger - Roger Rabbit - Guild of Thieves - ed altri.
Tel. 0322/53312 e chiedere di Davide (Novara).

Commodore 128, Regitratori, Monitor B/N, Duplicatore, Autofire, Penna ottica, 1000 giochi ca + Utility, Video Basic, ecc.,il tutto a 750.000 trattabili.
Attenzione!! Vendo anche ultime novità di giochi per C64.
Telefonare dalle 15.00 alle 20.30 allo 080/372252 - (BA)

Scambio programmi e manuali Amiga (o vendo a prezzi bassissimi es: L. 3.000 disco compreso). Ho oltre 1000 titoli, rispondo a chiunque.
Mauro Bricca - Via Monade n. 38 - 18013 Diano Marina (IM) - Tel. 0183/400814 o 0183/495491.

Cercasi disperatamente manuale Simons Basic e programmi per stampare (Print Shop - News Room ecc.)
Scrivere a Manganiello Gianluca - Via Viggiano n.3 - 83031 Ariano (IP)

Vendo software per Amiga 500 e 2000 a prezzi modici.
Eventualmente scambio.
Scrivere a Bettega Daniele - Via Pace n. 16 - 36034 Malo (VI) - Tel. 0445/602054
Annuncio sempre valido.

Vendo C64 nuovo solo 2 mesi + 30 cassette giochi ognuna con 10 giochi in media + Joy a Lire 400.000.
Concordia Luciano Tel. 0833/861754

Vendo, scambio, compro software per C64 su disco.
Cerco istruzioni per FS II, PSF e IFILTRATOR.
Tel. 045/917589

Cerco fotocopie delle istruzioni originali di Simon's Basic per C64 (è ovvio tutto a mie spese).
Romano Antonio - Pass. Ciaikowski n. 2 - 90145 (PA) - Tel. 091/205636

Compro, vendo, scambio!! giochi favolosi, programmi, manuali per Amiga.
Sono disposto anche a formare un Club.
Marco Vigoni - Via Cassiodoro n. 22 - (VA) - Tel. 0332/260367.
Annuncio sempre valido.

Permuto PRG - Utility e giochi su disco per C64.
Scrivere a Sebastiano Messina - Via Rossini n. 28 - 93017 San Cataldo (CL)

Vendiamo programmi su disco per CBM 64/128 a 3.000 l'uno (floppy escluso).
Disponiamo di circa 700 programmi, giochi, utilities, grafici, musicali con arrivi settimanali di nuovo software.
Massima serietà.
Roberto e Luca - Tel. 02/28404772 (ore pomeridiane)

Vendiamo giochi su cassettta per C64 a 2.500 l'uno. Spedire anche cassetta vergine per la duplicazione del gioco.
Scrivere a Maiorano Giordano - Via Pietro Mascagni n. 178 - 41058 Vignola (MO)

Vendo software per C64 su disco o cassetta, possibilità di abbonamenti.
Max serietà.
Per informazioni scrivere a Marinaro Angelo - Via G. Mateotti n. 7 - 86100 Campobasso oppure Via Liguria n. 19 - 56100 (PS)

Vendo C64 + registratore + disk drive 1541 + monitor + Ben compatibile (a schermo verde) + 10 dischetti con portadischi + 1 Joy + regitratore con 8 cassette.
Vendo tutto a 700.000 trattabili.
Telefonare allo 075/7512110 - 603888 (Perugia) nelle ore dei pasti.

Commodore 128+ monitor Hanterex + disk drive 1571 + joy stick + 15 F.D. Vendo tutto a 950.000.
Telefonare a Pietro 06/3600015

Ultime novità per Amiga prezzi ridicoli!!! Solo presso il S.A.C.
Telefonare al 0362/501857, dalle ore 20.00 alle 21.00 e chiedere di Luca.

Acquisto per Commodore C64 Floppy disk 1541/2 a 150.000.
Telefonare a Massimo la mattina 0766/734656

Vendo a blocco o separatamente sistema per sviluppo e stampa Totocalcio Totip Enalotto componente: C64/Disk Drive 1541/ MPS 802 Grafica/ Stampante E.13 per schedine/Programma Alfa Toto Sistem.
Telefonare o scrivere a: Massimiliano Testa - via Ix Maggio 92 - 93016 Riesi - Tel.0934/921138.

Vendo C64 + registratore + regalo di 10 cassette di videogames a sole 200.000.
Fornara Giorgio, Via Cialdini n.6 - 20022 Castano Primo (MI), Tel. 0331/881323.

L'Aquila Amiga Soft vende programmi a prezzi irrisori ultime novità.
Telefonare 0864/63381 (alla sera) e chiedere di David.

vendo giochi per Amiga 500, in abbonamento e singolarmente max serietà, ultime novità.
Giuseppe Malaspina - Via Matteotti - Tel. 0771/24618.

Vendo "Final Turbo IV" con Turboload, Freezer, Hard copy ad icone a 75.000.
Simone Avanzi - P.za Erbe n. 38 - 37121 verona - Tel. 045/30437.

Vendo e scambio numerosissimi manuali come - Sculpt 3D - Amiga Dos e tanti altri - per Amiga (ho tanto Software) - Redavid Luca - Via casini n. 4 -Tel. 051/511460 - 40127 (BO).

Vendiamo le ultime novità Software per C64 a 5.000 il dischetto, registrato interamente su entrambe le facciate e compreso il supporto magnetico. Rapide spedizioni in contrassegno in tuutt'Italia. Assicuriamo la massima serietà.
THE Anormals group - cas. p.n.50 -07037 Sorso (SS).
Tel. 079/350071/350393/ 351276 per informazioni.

Vendo e scambio programmi, giochi ecc. Dispongo di manuali introvabili tutto per Amiga.
redavid Luca - Via casini n.4 - 40127 (BO).
Tel. 051/511460

Sono un ragazzo che ha il C64 duplico giochi su disco a 900; con disco a 6.500.
Boldini Matteo - Via Bianchi Ferrari n.92 - (MD)
Tel. 059/352052

Vendo C128 + Drive 1571 + Monitor F.V.+ Stampante MPS 803 + 2 Joy + 40 Dischetti + 5 libri.
1.190.000 - Tel. 0884/21566

Disppongo ultime novità Amiga - Assicurata max serietà - telefonare allo 0376/619557 dopo le ore 19.00 e chiedere di sergio.

Acquisto per contanti Disk Drive 1541 - 1571, GEOS,MS-DOS GWBASIC per C64 (con manuali).
Telefonare allo 0363/988485 e chiedere di Carmelo.

Vendo riviste Commodore dei seguenti anni:
Anno (1984) n.1 - 2 - 3 - 4 Anno (1985) n. 7 - 10 - 11 - 13 - 14 - 15;
Commodore Computer Club:
Anno (1984) n. 13 - 15 Anno (1986) n. 36 Anno (1987) n. 38 - 39 - 40 - 41 Anno (1989) n. 60 - 61; il tutto a 17.500, per chi fosse interessato scrivere al seguente indirizzo: Calvano Carlo, Viale Europa n.104,91011 Alcano (TP)

Cerco manuale (anche fotocopia) del Macro Assembler in italiano.
Telefonare allo 091/6885475 e chiedere di Giuseppe

Cerco programmi per C128 in modalità CP/M e C128.
Scrivere a Tiziano Fortin, Via Omero n.6 - 20015 Parabiago (MI)

Scambio Software di ogni tipo su disco per C64.
Cerco le istruzioni del GEOS e del 64 MANAGER.
Goffredo Vito - Viale Ionio -74025 Marina di Ginosa (TA) -Tel. 099/627403

Fornisco metodi per il gioco del LOTTO del valore di oltre L. 2.000.000 in cambio di Computer medio usato provvisto di Drive.
Telefonare allo 08302/747407 - Squinzano (Lecce)

Vendesi C/128, driver 1571 compatibile, monitor B/N Philips, Stampante MPS 803 - compatibile
Vanini Angelo - Telefonare dalle 18.00 alle 21.00 al 051/932385

Vendo C64, Registratore, Joystick, circa 300 progr. per L. 300.000 trattabili.
Tel 071/894352 (ore pasti)

Cambio - vendo programmi per Amiga sempre in rinnovo tra questi Dragon's Line - Italy 90 - Roger Rabbit ecc.
Telefona allo 081/5544352 - chiedere di Riccardo.

Cerco: Espansione 32K per TI 99 4/A - Ext Basic ed eventuali Hardware e software su sss o cassetta.
Rossi Riccardo - Via Rivalivenza n. 17 - Motta di Livenza.
Tel. 0422/768640.

Cerco per Spectravideo SV328 programmi su cassetta, eventuale stampante ed altro Hardware.
Cerco anche programmi per Comx35
Rossi Riccardo - Via Rivalivenza n.17 - Motta di Livenza - Tel. 0422/768640

Eccezionale!! (Tutto per il C64), scambio i giochi: Ikary Warriors, Pal Land, Calcio, Basket, Tager Renage, Pitstop II (tutto su una cassetta).
Per il Gw-Basic e Ms-Dos (versione 2) su cassetta + manuqale d'istruzioni valido solo per provincia di Vicenza; max serietà.
Tel. 0445/370976 (chiedere di Alberto e telefonare dalle 20.00 alle 20.30) - Via S. Paolo n. 12 - 36016 Thiene (VI)

Vendo C 128 + registratore + 300 giochi + 60 programmi di Utilities + Joy, il tutto a 500.000.
Lauriola Salvatore - Via Sacini n. 43 - (BA)

Vendo per C64 (disco) stupendi giochi come: Rocket Ranger, R-Tipe, Robocop, Ugh - Lympics, Microsoccer e molti altri. Novità sempre in arrivo!!!
Marco Zanelli - Via Ardaro n. 38 - 38066 Riva (TN) -Tel. 0464/ 551714

L'Amiga Entertainment Club cerca nuovi Amigos.
Bollettino mensile, P.D. Software, iniziative varie, scambio di esperienze, consigli e programmi.
Ai nuovi soci in omaggio uno splendido Demo.
Dunque cosa aspettate?..... Per qualsiasi informazione scrivete pure a Canestrale Vincenzo - Via Fini n. 21 - 71100 Foggia.

Scambio programmi C64.
Tel. 02/2545465 Beppe
Cerco arredare (Systems)
Barbian II - Stifflip & CO.
Offro Micro Prose Soccer
Captain Blood -Sky Travel

Vi annuncio della fondazione di un nuovo Computer Club, il Computer Friends; per maggiori informazioni telefonare o scrivere al seguente indirizzo: Fabrizio Scaturro - Via XXI n. 1 - 91

Cerco giochi o utility per C 128 (solo modo 128). Inviatemi le vostre liste.
Andrea Grazini - Via Lodovico Moro n. 61 - Tel. 02/476434.

Vendo e Scambio programmi per C64 su disco. Ho diversi programmi su 300 dischi tra nuovi e vecchi.
Scrivere solamente inviando vostra lista a Marco Riazzola - Via G. Gaslini n. 2 - 20052 Monza (MI).
Inoltre cerco molti programmi grafici su disco.

BOY SOFT
Videogiochi su disco per C64 (5 1/2) di tutti i tipi a 2.000 ca.
Luca - Tel 06/8180303 -(RO)

Vendo giochi per CBM 64, richiedere listino con 2 Bolli da 650.
Cimmino Danilo - C.so Mediterraneo n. 10 - 80016 Marano (NA) - Tel. 7428463

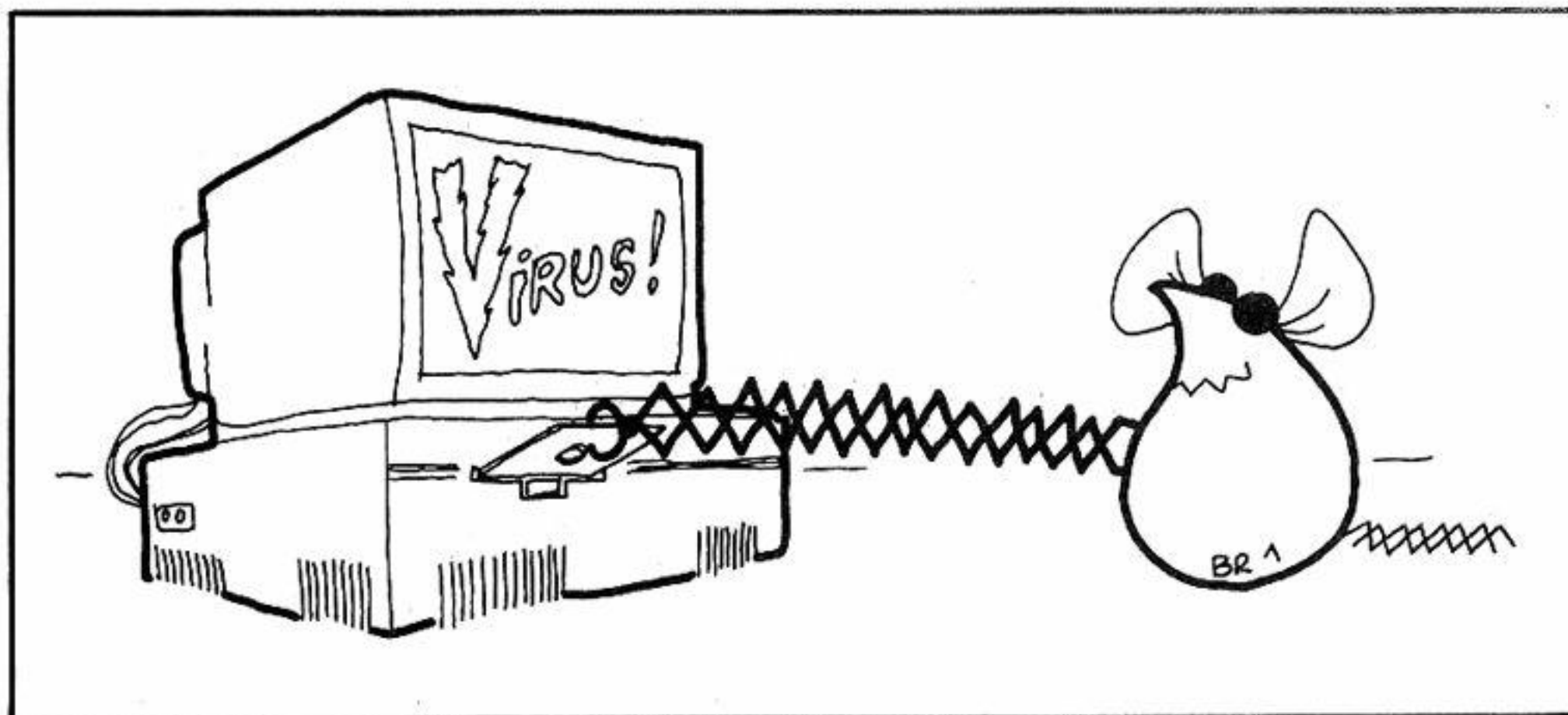
Vendita Soft per Amiga C64... Novità settimanali provenienti anche dall'estero elaborazionni grafico - sonore - massima serietà - vendita postale.
Tonisoft Tel. 085/74561 - Pescara.

Per errore ho rovinato il dischetto superfloppy 64 n.9 (calcio Toto Sprint) chi me lo fa copiare?
telefonare dalle ore 13.00 alle ore 20.00 allo 011/2481497.

Per C 128 i testi: Mapping the C 128; Tricks and tips; 1571 Internals; L.M. for Beginners, vendo anche singolarmente, Telefona ore pasti, chiedere di Carmine allo 081/7014031

Cerco in Torino, una copia funzionante del "Defender of the Crown " - Scambio con sue istruzioni.
Telefonare dopo le 20.00 al 343354
Giuseppe Rendina - Via G. Verga n.10 - 10135 (TO)
(Grazie)

Acquisto Disk Drive 1541 raccomando max serietà.
Lancellotti Alfredo - Vico Castellina alla Stella n. 9 -80137 (NA) - Tel. 081/294596

Vendo programmi per Amiga, ultime novità a prezzo ridicolo.
Per informazioni Telefonare allo 0362/501857 dalle ore 20.00 alle 21.00. (chiedere di Luca)

UN ANTIVIRUS PER CHI INIZIA

Non tutti i lettori sanno che cosa sia un virus; e che si può combatterlo anche con un breve programma in Basic

di **Claudio Bertone**

Con il presente articolo non solo intendiamo offrire una breve panoramica sui virus che affliggono l'Amiga ma, soprattutto, proporre un vaccino per eliminarli in modo semplice.

I programmi di queste pagine sono due: uno è in Basic, e il suo utilizzo può essere effettuato anche dai meno esperti; l'altro, in C, è invece più complesso a causa delle routines di accesso al disco.

Per costituire un'efficace arma contro i virus, tuttavia, i programmi dovrebbero essere usati entrambi.

COME FUNZIONA L'ANTIVIRUS

Il programma scritto in C permette di ridurre a zero i rischi; esso presenta, sullo schermo, il contenuto del Boot Block dei dischi. Con un po' di pratica, infatti, sarà possibile distinguere quelli "buoni" da quelli "cattivi" poichè generalmente tutti i virus contengono frasi facilmente riconoscibili.

Il suo funzionamento, come detto precedentemente, non è semplicissimo; ci limiteremo a dire che legge il contenuto dei blocchi 0 e 1 (Boot Block), trascrivendolo sullo schermo in caratteri ASCII.

Il programma richiede un solo parametro, che si riferisce al numero del drive che si vuole analizzare. Per esempio, digitando "VIRUS 0" verrà esaminato il contenuto del disco del drive interno.

Il programma in Basic, invece, è nato per sopperire ad un difetto del programma precedente. Esso è composto da due schermate che hanno compiti diversi. Quella realmente efficace è la seconda; la prima ha un compito essenzialmente visivo; serve a mostrare, all'utente inesperto, l'eventuale presenza del virus SCA in memoria, il quale si alloca a partire dalla locazione esadecimale 7EC00, per una lunghezza di 1K.

La seconda videata, invece, serve nei casi in cui, con il programma in C prima citato, sia stato in precedenza visualizzato un Boot Block sospetto, cioè privo di messaggi che lo identifichino come virus, ma non simile a quello standard.

Basterà, in questi casi, resettare l'Amiga, far configurare il sistema con il disco sospetto e lanciare il programma Basic (da un dischetto, ovviamente, protetto contro la scrittura). Se verranno mostrate anoma-

lie nella struttura ExecBase, si tratterà con molta probabilità di un virus.

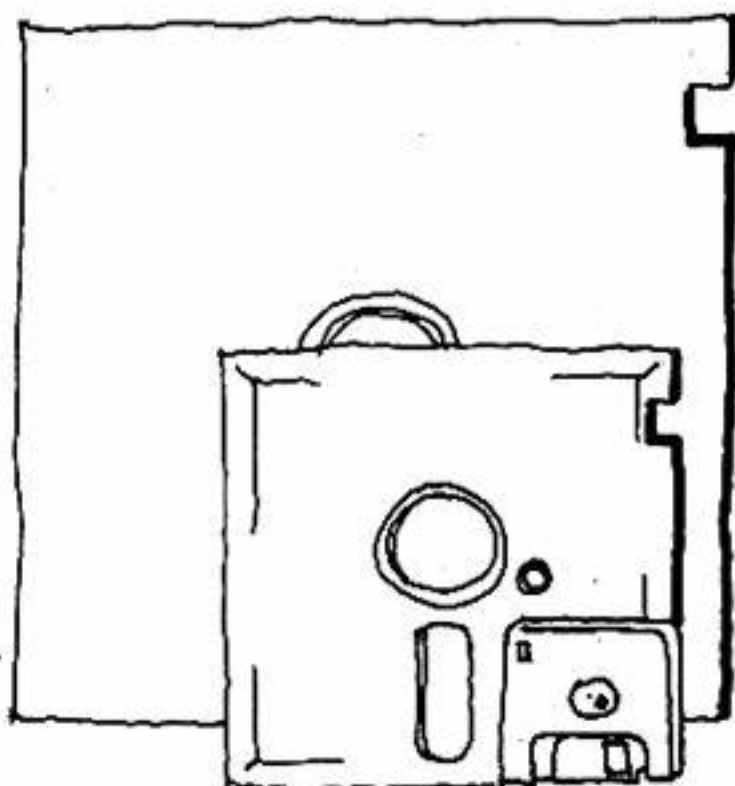
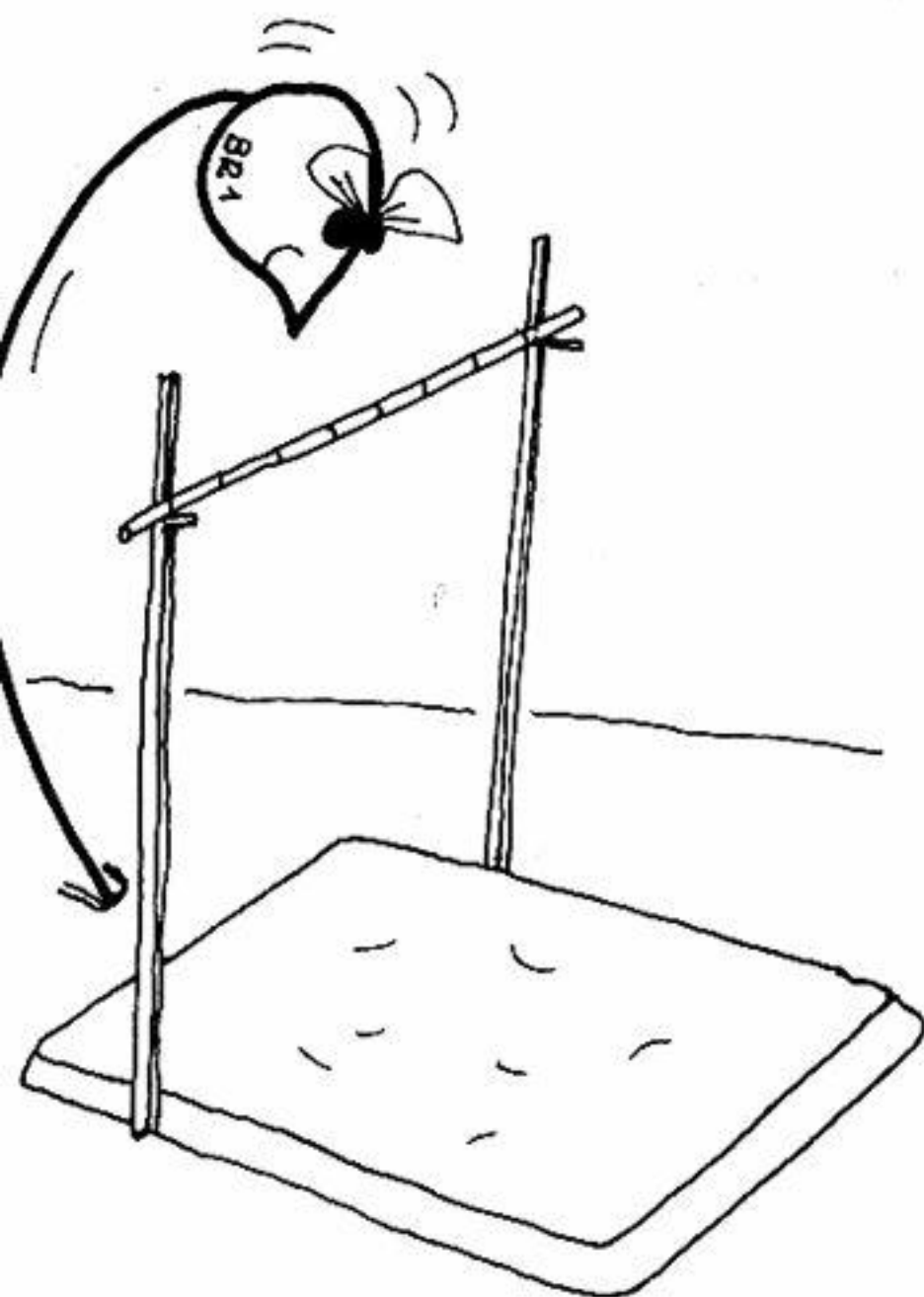
Lo si potrà dunque eliminare senza pietà con il comando Install. Se volete essere proprio sicuri di non danneggiare irreparabilmente un programma a cui tenete, vi consigliamo di effettuare una copia del disco sospetto, attivare Install sulla copia e vedere se questa, in seguito, si carica lo stesso. Se quest'ultima operazione fallisce vuol dire che il Boot Block era un caricatore di programmi (o una protezione vitale); altrimenti era un virus.

Ricordatevi, una volta che avete riscritto sul disco il Boot Block corretto, di posizionare la linguetta di protezione contro la scrittura, in modo che non possa più essere contagiato.

COME FUNZIONA

Il funzionamento del programma Basic è semplice: si preleva il puntatore alla struttura ExecBase dalla locazione 4 e, con alcuni calcoli, si accede ai vari vettori che contiene, verificando se sono settati ai valori giusti e correggendo, dopo un'opportuna segnalazione all'utente, le eventuali difformità.

I programmi, in verità, sono validi solo contro il virus SCA, con il quale la tecnica descritta funziona egregiamente. Se ritenete di possedere dischetti infettati da virus che vi danno filo da torcere (e non riuscite a "curarli") inviatene una copia in Redazione. Provvederemo a scrivere altri programmi antivirus efficaci; e, naturalmente, a pubblicarne i listati corrispondenti.

U F O

COME AGISCONO I VIRUS

I lettori che appena acquistato un Amiga forse non sanno che cosa siano gli "agenti patogeni" che devono esser combattuti dagli utenti di questo straordinario computer.

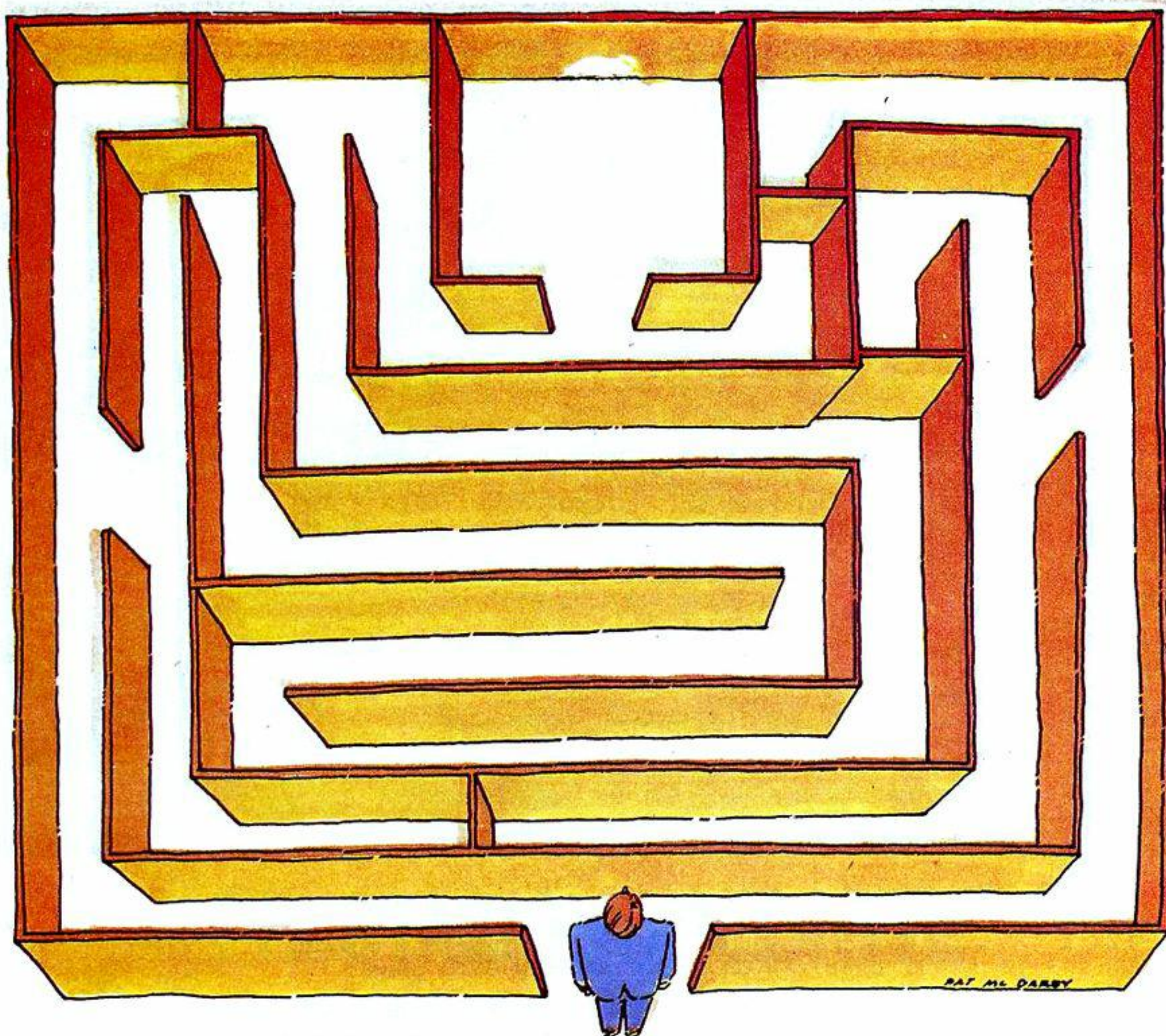
Attualmente, per l'Amiga, sono diffusi (ed attivissimi) numerosi virus: alcuni di essi sono nocivi, come il tristemente noto "Byte Bandit" che provoca danni sui dischi nei quali riesce ad innestarsi; altri, fortunatamente, si limitano a far apparire sullo schermo messaggi di vario tipo (come il virus "SCA"). Per fortuna hanno tutti un punto debole in comune, che è quello di dover necessariamente risiedere sul Boot Block del disco. Spieghiamoci meglio.

Quando viene eseguita la configurazione del sistema, il computer passa automaticamente, ed inconsapevolmente, il controllo al virus (se, ovviamente, presente), il quale provvede ad installarsi nella parte alta della memoria in modo da rendersi immune al reset dal momento che modifica alcuni vettori della struttura ExecBase. Inoltre il virus provvede, seguendo tecniche che possono esser diverse tra loro, ad inserirsi all'interno di un ciclo di interrupt, in modo da "contagiare" facilmente altri dischetti.

Sembrerebbe che, per esser sicuri, sia sufficiente attivare il comando Install (presente in CLI) prima di inserire un qualsiasi disco "sospetto" nel drive del computer.

L'errore che però si commette, attivando da CLI il comando Install, può esser grave poichè il comando provvede a distruggere i dati allocati nel Boot Block, indipendentemente dal suo contenuto. Se, infatti, invece del virus è presente, ad esempio, il caricatore veloce di un programma o un gruppo di dati che fanno parte di un "innocuo" sistema di protezione (caso molto frequente, soprattutto nei programmi commerciali), il dischetto non sarà più utilizzabile.

```
' ANTIVIRUS   v1.0
'
' di BERTONE CLAUDIO

DIM n$(15),ind&(15),va&(15)

ind&=PEEKL (4)   ' Puntatore alla struttura ExecBase
alt&=7*65536&+14*4096+12*256 'Punta alla parte alta della memoria (#$7ec00)
t%=0 : x%=0 'Variabili di comodo
st$="" 'Contiene i caratteri ascii del blocco da 7ec00 a 7efff

FOR t%=0 TO 4 :READ n$(t%),va&(t%) :ind&(t%)=ind&+42+t%*4 :NEXT t%
FOR t%=5 TO 8 :READ n$(t%),va&(t%) :ind&(t%)=ind&+66+(t%-5)*4 :NEXT t%
FOR t%=9 TO 12 :READ n$(t%),va&(t%) :ind&(t%)=ind&+300+(t%-9)*4 :NEXT t%
FOR t%=13 TO 15 :READ n$(t%),va&(t%) :ind&(t%)=ind&+546+(t%-13)*4 :NEXT t%

CLS:PRINT "Esame della memoria da #$7ec00 a #$7efff:"
PRINT:FOR t%=0 TO 1023
x%=PEEK (alt&+t%): IF x%<32 THEN x%=ASC(".")
IF x%>128 THEN x%=ASC(".")
st$=st$+CHR$(x%):NEXT
FOR t%=0 TO 19
PRINT "     ";
PRINT MID$(st$,1+t%*64,64)
NEXT:PRINT "PREMI <RETURN> PER IL SECONDO TEST";
IF INSTR(st$,"SCA")<>0 AND INSTR(st$,"infected")<>0 THEN
LOCATE 1,42:PRINT "Il VIRUS della SCA e' in azione !!"
ELSE
LOCATE 1,42:PRINT "In memoria non ci sono virus !!"
END IF
a$="":WHILE a$<>CHR$(13):a$=INKEY$:WEND
x%=0
CLS:PRINT "Esame della struttura ExecBase:":PRINT
FOR t%=0 TO 15
IF PEEKL (ind&(t%)) <> va&(t%) THEN
PRINT "- Il vettore "n$(t%)" e' stato modificato !!"
POKEL ind&(t%),va&(t%):x%=1
END IF
NEXT
IF x%=0 THEN PRINT:PRINT "La struttura ExecBase e' a posto !":END
PRINT "La struttura ExecBase e' stata settata ai valori normali":PRINT
PRINT "ATTENZIONE: e' comunque MOLTO probabile che sul disco dal"
PRINT "            quale e' stato effettuato il Boot del sistema"
PRINT "            sia presente un VIRUS. Andate in ambiente CLI"
PRINT "            e digitate, dopo aver inserito il disco di"
PRINT "            sistema: 'INSTALL DRIVE DF0:'."
PRINT:END

DATA ColdCapture,0,CoolCapture,0,WarmCapture,0
DATA SysStkUpper,524288,SysStkLower,518144
DATA DebugEntry,16524098,DebugData,0,AlertData,0
DATA MaxExtMem,0,ResModules,7168,TaskTrapCode,16527284
DATA TaskExceptCode,16527284,TaskExitCode,16522476
DATA KickMemPtr,0,KickTagPtr,0,KickCheckSum,0
```

```
#include <exec/types.h>
#include <exec/io.h>
#include <devices/trackdisk.h>
#define BLOCK 1024
struct Port *dport;
struct IOStdReq *dreq;
extern struct MsgPort *CreatePort ();
extern struct IOStdReq *CreateStdIO ();
BYTE bootblock [BLOCK];
main (argc,argv)
int argc;
char *argv[];
{
int flag, x, y;
printf ("\n\n\033[33;1m                VIRUS\n\n");
printf ("\033[31m    di BERTONE CLAUDIO\n\033[0m");
printf ("                          \n");
printf ("                            \n\n");
if (argc != 2) {
printf ("\nUso: Virus numero_drive ( 0,1,2,3,4 ... )\n\n") ;
exit(0);
}
/* Preparo l'accesso al disco */
dport=CreatePort(0,0);
dreq =CreateStdIO(dport);
if ((flag = OpenDevice (TD_NAME,atoi (argv[1]),dreq,0)) != 0) {
printf ("\nBOOT: Non riesco ad aprire il drive  DF%d:\n\n",atoi(argv[1]));
exit(5);
}
printf ("\n\n        Visione in caratteri ASCII del BootBlock:\n");
/* Accendo il motore */
dreq->io_Length =1;
dreq->io_Command= TD_MOTOR;
DoIO(dreq);
/* Accedo alle Key 0 e 1 */
dreq->io_Length = BLOCK;
dreq->io_Data   = &bootblock[0];
dreq->io_Command= CMD_READ;
dreq->io_Offset = 0;
DoIO(dreq);
/* Rappresentazione in caratteri ASCII */
for (x=0; x<16; x++) {
printf ("\n ");
for (y=0; y<64; y++)
putchar (((flag=bootblock[y+x*64]) >31 && flag<128 ) ? flag : '.');
}
printf ("\n");
/* Spengo il motore */
dreq->io_Length =0;
dreq->io_Command= TD_MOTOR;
DoIO (dreq);
CloseDevice(dreq);
DeleteStdIO(dreq);
DeletePcrt(dport);
}
```

UN DIABOLICO LABIRINTO

Un micro-gioco inviato da un nostro lettore vi vedrà impegnati contro il computer

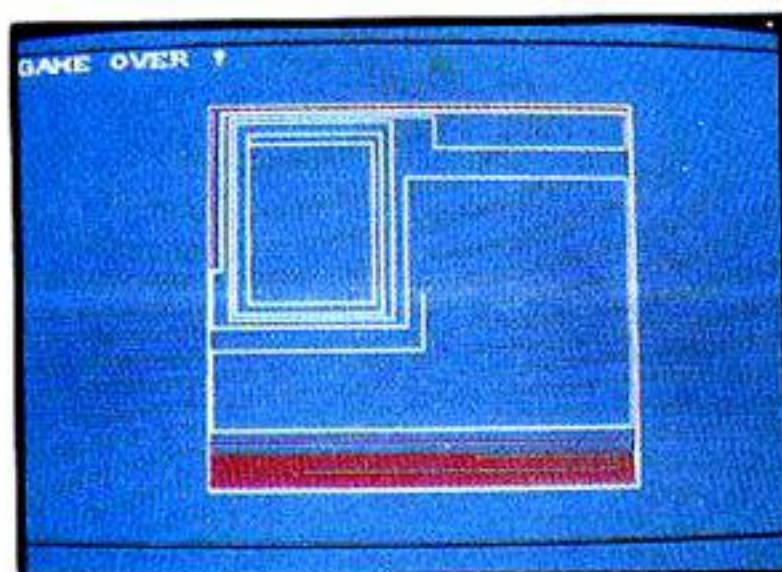
di **Claudio Bertone**

Il programma di queste pagine è la dimostrazione pratica di come si possa realizzare, con un pochissime righe Basic, un gioco divertente.

Si deve guidare un puntino luminoso (pixel) che lascia una scia dietro di sè. Non bisogna passare nè sulla scia lasciata dal proprio pixel, nè su quelle dei pixel tracciati dal computer. Si deve inoltre tentare di battere il computer costringendolo a passare su una delle tracce.

Il programma consta di 29(!) livelli ad ognuno dei quali corrisponde la quantità di scie tracciate contemporaneamente da Amiga.

Il funzionamento prevede l'inserimento del Joystick in porta 2 (il mouse, quindi, continua a svolgere le sue funzioni). Per interrompere il gioco è sufficiente premere Ctrl e "C" (o usare il mouse: attenzione ai colori!). Per riprendere a giocare, dopo il fatidico "Game Over", bisogna premere un tasto qualsiasi.

```
' *********** TRON ************
' Creato da BERTONE CLAUDIO per C.C.C.
SCREEN 1,320,200,5,1:WINDOW 2,,,0,1:RANDOMIZE TIMER:DEFINT a-z
DIM SHARED x(31),dx(31),y(31),dy(31)
PALETTE 0,0,0,0:PALETTE 1,0,0,0:PALETTE 2,1,1,1
FOR t=3 TO 31:PALETTE t,1-t/31,t/31,RND(1):NEXT:COLOR 2
main:
CLS:LOCATE 1,1:PRINT "  LIVELLO:":x=240:y=175:LINE (80,25)-(x,y),2,b
FOR nu=3 TO 31:x(2)=160:y(2)=100:dx(2)=0:dy(2)=1:FOR t=3 TO nu
x(t)=INT(RND (1) * (x-81)) +81:y(t)=INT(RND (1) * (y-26))
+26:dx(t)=INT(RND(1)*3-1)
IF dx(t)=0 THEN dy(t)=INT(RND (1)*3)-1:IF dy(t)=0 THEN dy(t)=1
IF dx(t)<>0 THEN dy(t)=0
NEXT t:LOCATE 1,11:PRINT nu-2:LINE (81,26)-(x-1,y-1),0,bf:GOSUB game
IF n=2 THEN BEEP:LOCATE 1,1:PRINT "GAME OVER !  ":CALL Premi:GOTO main
BEEP:NEXT:CLS:WINDOW CLOSE 2:SCREEN CLOSE 1
PRINT "COMPLIMENTI:":PRINT:PRINT "HAI SUPERATO TUTTI I 29"
PRINT "LIVELLI DEL GIOCO.":PRINT "SEI IMBATTIBILE !!!":END
game:
n=2:f=STICK (2) :IF f<>0 AND dx(2)=0 THEN dx(2)=f:dy(2)=0
f=STICK (3) :IF f<>0 AND dy(2)=0 THEN dy(2)=f:dx(2)=0
CALL Prova (n): IF n<>-1 THEN RETURN
FOR t=3 TO nu:n=t:IF dx(t)=0 AND dy(t)=0 THEN GOTO exnext
CALL Prova (n):IF n=-1 GOTO exnext
IF dx(t)=0 THEN
dx(t)=1:dy(t)=0:CALL Prova (n):IF n=-1 GOTO exif
dx(t)=-1:CALL Prova (n)
ELSE
dy(t)=1:dx(t)=0:CALL Prova (n): IF n=-1 GOTO exif
dy(t)=-1:CALL Prova(n)
exif:
END IF
IF n<>-1 THEN dx(t)=0:dy(t)=0
exnext:
NEXT t
FOR t=3 TO nu :IF dx(t)<>0 OR dy(t)<>0 THEN GOTO game
NEXT
n=-1
RETURN
SUB Prova (n) STATIC
IF POINT (x(n)+dx(n),y(n)+dy(n))=0 THEN x(n)=x(n)+dx(n):y(n)=y(n)+dy(n):PSET
(x(n),y(n)),n:n=-1
END SUB
SUB Premi STATIC
ke$="":WHILE ke$="":ke$=INKEY$:WEND
END SUB
```

PRESTAZIONI RELATIVE

Quando si tratta di gestire grandi quantità di dati, sembra inevitabile il ricorso a potenti e veloci elaboratori. Eppure, con un semplice C/128 o C/64...

di **Gregor Samsa**

Chiunque possieda un disk drive, anche se solo da poco tempo, ha modo di familiarizzare in fretta con certe sigle che caratterizzano la directory di ogni dischetto.

Anzi, col passare del tempo, certe associazioni mentali diventano quasi automatiche:

Questo è un file Prg, e quindi lo posso caricare direttamente. Quello, invece, siglato Seq, è il frutto di un elaboratore di testi oppure è l'archivio di qualcosa.

In qualche rara occasione può capitare persino di vedere un file *Usr* o addirittura *Del* (sicuro intervento di cracker) ma, pensateci un po' su, quante volte vi siete imbattuti nella fantomatica sigla *Rel*?

Eppure la possibilità di sfruttare files relativi è forse una delle più potenti risorse dei piccoli di casa Commodore, tale da non far rimpiangere le prestazioni di Personal ben più quotati.

Lo scarso uso che se ne fa è probabilmente legato ad una certa difficoltà di utilizzo, in realtà più presunta che reale.

Se di un certo livello, anche la memorizzazione di dati in un file sequenziale richiede infatti una certa accuratezza, peraltro non ripagata dalla flessibilità propria dei files relativi.

Ciò non vuol dire, ovviamente, che questi ultimi debbano essere impiegati in ogni occasione, ma solo che, in tema di archiviazioni, dovrebbero rappresentare la risorsa di elezione.

APPRENDISTATO

Prima di entrare nello specifico, è necessario avere ben chiare le idee sulla struttura *logica* di un sistema di memorizzazione dati, o Database che dir si voglia, e sulla terminologia ad esso associata.

In pratica, si tratta semplicemente di scindere una unità superiore, il file, nelle

SCELTE OBBLIGATE

A dispetto della struttura *logica* di un file-archivio, di fatto i dati in esso contenuti sono organizzati in sequenza, più o meno come illustrato nella parte bassa di figura 1.

Generalizzando, si può dire che tale struttura non cambia quale che sia il tipo di file adoperato ma, a seconda della sua gestione, l'accesso ai dati può assumere connotazioni molto differenti, soprattutto in relazione alla velocità.

Chiariamo meglio.

Supponiamo che il file mostrato in figura sia di tipo sequenziale, la forma certamente a tutti più nota.

In questo caso, della divisione in Record e Campi, al Dos non importa nulla: tutto ciò che bisogna fornirgli, è una serie di istruzioni *Input#* o *Get#*: "lui" restituirà il dato.

Volendo, per esempio, accedere al *secondo* cognome, che sappiamo essere il primo campo del *secondo* record, dovremo necessariamente effettuare quattro operazioni: leggere, anche se solo per scartarli, i primi tre campi, ed infine leggere il dato che interessa.

I più esperti obietteranno che caricando, prima, il file in memoria, questo tipo di operazioni non crea problemi di velocità.

Ciò è vero, ma si provi ad immaginare un file composto da 500 record, ognuno comprendente 10 campi.

Disponibilità di memoria Ram a parte, certamente non da scialo nel caso di un C/64, per accedere al primo campo del record 400, sarebbero necessarie la bellezza di 4000 operazioni in lettura!

Il metodo sequenziale, in questi casi, mostra decisamente i suoi limiti, rendendo quasi indispensabile il ricorso all'accesso casuale.

In questo caso, infatti, pur agendo sulla stessa struttura fisica, si potrà leggere, direttamente, il record 400 e, all'interno di questo, il suo campo numero 10.

subunità descrittive che lo compongono: record, campi, e singoli caratteri.

Come illustrato nella parte alta di figura 1, è un po' come giocare con le scatole cinesi: un file è diviso in record, un record in campi (*field*), un campo in singoli caratteri (o *byte*).

Più in concreto, immaginiamo un classico indirizzario, contenuto nella nostra agenda (il file), per brevità... senza indirizzi (che razza di indirizzario è?), ma con solo cognome, nome e città di residenza.

Ogni gruppo di informazione tipo...

Rossi Mario - Milano

...costituisce un Record, mentre le singole voci Rossi, Mario e Milano rappresentano i campi.

In definitiva, ci troviamo di fronte ad un file (il foglio dell'agenda) e ad un certo numero di record, ognuno dei quali è composto da tre campi (cognome, nome e città).

Se tutto è chiaro, non resta che gettare (metaforicamente) alle ortiche l'agenda: è il momento di tirar fuori i nostri amati floppy.

QUANDO VUOI, DOVE VUOI

Ciò che vogliamo ottenere, in parole povere, è la possibilità di accedere direttamente a qualunque record del file, tanto in scrittura che in lettura.

Un simile vantaggio, rispetto all'accesso sequenziale, non solo consente una buona velocità, ma, soprattutto, pone come unico limite, alla quantità di dati archiviabili, la capienza fisica del dischetto.

Per disporre di tale facilitazione, tutto ciò che bisogna fare è ricorrere ai files relativi.

In fondo, soprattutto con il potente basic 7.0 del C/128, la loro gestione non è poi così complessa, in quanto la maggior parte del lavoro la svolge il computer e la sua diletta periferica, il disk drive.

A noi semplici opportunisti serve solo sapere che è necessario:

1) Aprire e *preparare* un file relativo.
2) Comunicare su quale numero di record si vuole operare.
3) Leggere o scrivere le informazioni che interessano.
4) Chiudere il file.

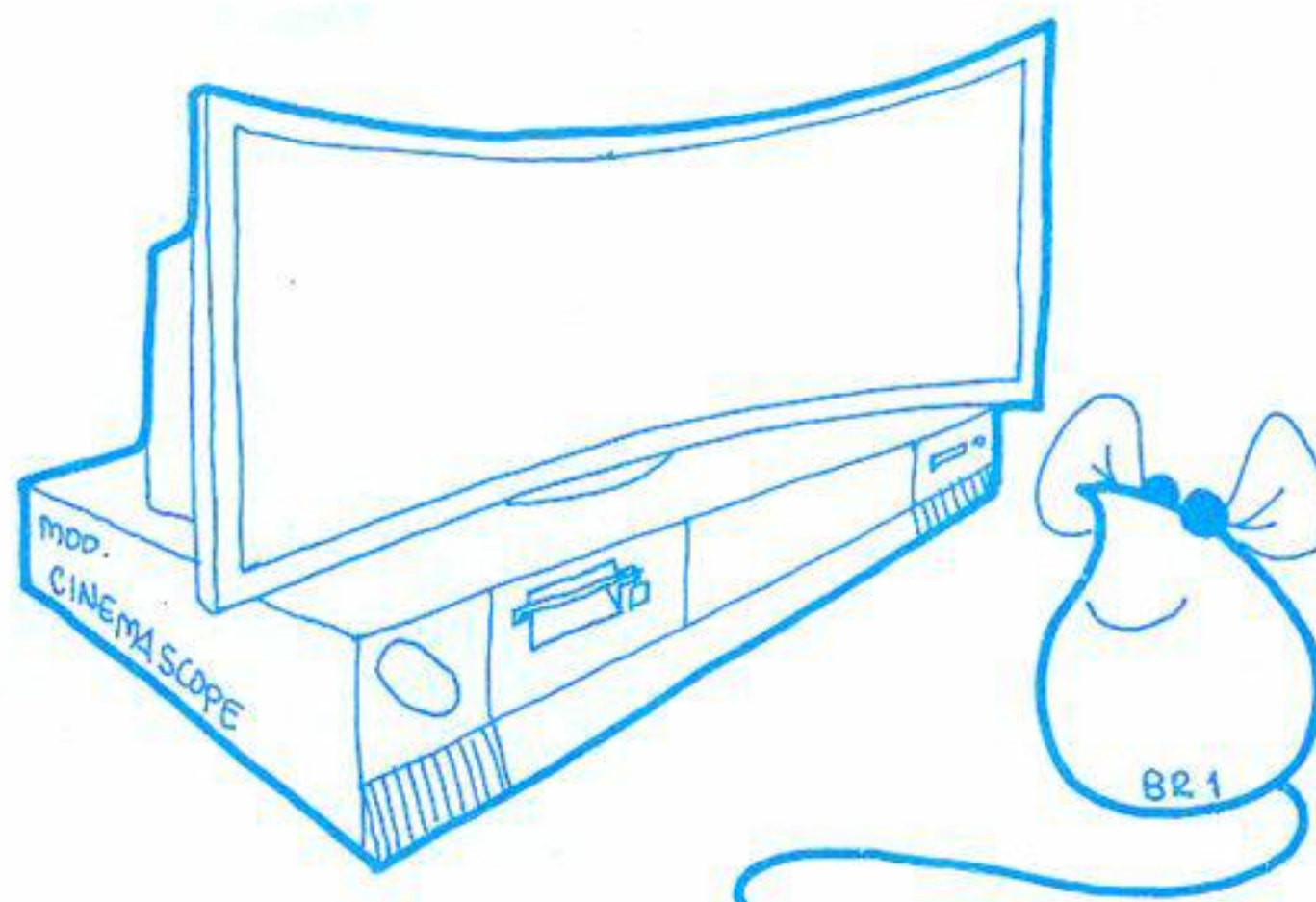
Penserà il sistema, in maniera automatica, ad integrare le precedenti operazioni con altre che comprendono, per esempio, la creazione su disco del *Side Sector*, una specie di indice per rintracciare fisicamente i record nel dischetto.

Ma vediamo di approfondire le fasi appena citate, chiaramente distinguibili anche nel programma listato in queste pagine (se ne parlerà tra breve).

L'apertura del file prevede anche una determinazione preventiva della dimensione di ogni record, che non può superare i 254 caratteri, o byte che dir si voglia.

Quindi, rimandando al manuale del drive per maggiori chiarimenti sulla sintassi, per il C/128 basterà...

Dopen #5, "nomefile", L 200

...per aprire un file relativo con record di 200 caratteri, mentre per impartire lo stesso comando sul C/64, si dovrà usare la forma...

```
VARIABILI STRATEGICHE
----------------------------------------
DI LIBERA RIASSEGNAZIONE
----------------------------------------
R$      = Nome del file relativo.
K$      = Nome del file indice (seq).
R       = Numero massimo di record
D       = Dimensione di ogni record.
NC      = Numero di campi.
CP$()   = Intestazione dei campi.
LN()    = Max. lunghezza di ogni campo.
----------------------------------------
DI PROGRAMMA
----------------------------------------
MX      = Numero record archiviati.
KE$()   = Campo chiave (in memoria).
AR$     = Assomma tutti i campi.
AP      = File aperto/non aperto (1/0).
PP      = Archivio pieno (1).
```

Open 5, 8, 5, "nomefile, L," + Chr$ (200)

...dopo avere aperto anche il canale-comandi con il classico *Open 15, 8, 15* (solo per il C/64!).

Per tutte le operazioni successive, che l'apertura consente sia in lettura che in scrittura, basterà prima far *puntare* il sistema al record voluto.

Il che, tradotto in basic 7.0, e riferito al file appena aperto come esempio, diventa...

Record #5, numerorecord, numerobyte

L'ultimo parametro, non obbligatorio, specifica la posizione all'interno del Record, e, come descritto nel manuale del drive 1570/71, occorre impartire il comando *Record*#12 due volte, prima e dopo l'accesso al record stesso.

Stesso discorso vale per il C/64, il cui striminzito interprete Basic, però, non prevede alcun comando specifico; per puntare al record voluto, sarà necessario ricorrere alla (più complicata) procedura:

1) Dividere il numero di record nel suo byte alto (Hi) e byte basso (Low). In pratica, bisognerà applicare la formula...
HI = Int (record / 256)
LOW = record - hi * 256

2) Inviare sul canale comandi (15) una stringa così impostata:
Print #15, "P" + Chr$(5) + Chr$(low) + Chr$(Hi) + Chr$(byte)

Anche qui l'ultimo parametro si riferisce alla posizione all'interno del record, mentre il Chr$(5) fa riferimento al numero di file prima aperto come relativo.

A questo punto disponiamo di tutte le conoscenze di base per operare su un file relativo, dal momento che le vere e proprie operazioni di scrittura e lettura non si differenziano sintatticamente da quelle normalmente usate per i file sequenziali (*Print#, Get#, Input#*), come pure quelli adibiti alla chiusura del file (*Dclose# 5* per il C/128 e *Close* 512 per il C/64)

NON SOLO TEORIA

Come sempre, però, tra la teoria e la pratica c'è una bella differenza.

Vediamo dunque come va realmente affrontato un archivio basato sui file relativi, descrivendone più dettagliatamente i suoi punti cardine.

L'invito, prima di cominciare, è quello di approfondire quanto trattato riferendosi al programma di queste pagine, che applica in maniera didattica l'argomento, ovvero suddividendone le varie fasi in compartimenti stagni ben delimitati dalle righe Rem.

Una volta comprese a fondo le caratteristiche dell'accesso casuale, lo stesso potrà essere facilmente migliorato nella sua efficienza, o addirittura modificato nell'algoritmo di gestione, semplicemente perfezionando qualcuno dei suoi moduli, o aggiungendone di nuovi (vedi riquadro).

Ma procediamo con ordine.

Il programma, del quale ne viene fornita una versione per C/128 ed una per C/64, consente di creare, e gestire, un archivio relativo di qualunque genere.

Così com'è, "tratta" semplicemente una rubrica telefonica comprendente nome, cognome, prefisso e telefono, ma è strutturato in modo da rendere facoltative queste assegnazioni di campo.

Basta, infatti, modificare le variabili delle primissime righe (elencate e descritte anche nel riquadro a parte), per adattarlo a qualsiasi esigenza.

In pratica, se si desidera anche un campo *indirizzi*, basterà aggiungere una unità ad NC, sommare a D il numero di caratteri + 1 del nuovo campo, ed assegnare a CP$(5) ed LN(5) il nome e la lunghezza massima del campo.

Analogamente, si può incrementare il numero di record voluti, fissato dalla variabile R, fino a 700, limite massimo accettato dal sistema.

Come già detto, ogni singolo record non può superare però i 254 caratteri (indipendentemente dal numero di campi), ed inoltre bisogna tener presente la capienza del dischetto.

Supponendo, infatti, di progettare un archivio con 700 record di 254 byte, ognuno di essi corrisponderà fisicamente ad un settore del disco, per un totale di 700 blocchi.

La cosa non porrà problemi se si dispone di un C/128 con drive 1571, ma, con un più modesto 1541, al di là dei classici 664 blocchi non si può proprio andare.

Inoltre bisogna anche valutare la presenza del *Side Sector* che, anche se "invisibile", può occupare fino a sei blocchi del dischetto; ed ancora, come vedremo tra breve, la presenza di (almemo) un file indice.

Quindi, regola prima: fare bene i conti, prima di "esagerare" con le dimensioni di un archivio.

Sulla base di queste nuove precisazioni, torniamo ora al nostro programma, dando per scontato che sia già stato copiato e debitamente salvato su disco.

TUTTI IN FILE

Dopo il lancio, si può notare come appaia un menu di sole due voci, apparentemente di significato simile.

Il fatto è che, quando ancora non esiste il file relativo, non basta una semplice apertura per averlo immediatamente disponibile.

Occorre anche, usando un termine cor-

rente, "preestenderlo", ovvero creare a priori tutti i record che desideriamo utilizzare, ovviamente vuoti (almeno all'inizio).

Per far ciò non è necessario scriverli tutti: basta posizionare il puntatore sull' ultimo record (quello di numero più alto possibile), e depositarvi qualcosa con una normalissima *Print#*.

Così facendo il sistema è costretto a creare anche tutti i record che precedono l'ultimo, riempiendoli con i codici Ascii 255 nel primo byte e 0 negli altri.

E' questo, tra l'altro, il motivo dell'attesa (a volte lunghissima) necessaria quando si crea un archivio ex novo.

Tale tecnica consente non solo di avere, in seguito, una notevole velocità di accesso, ma anche di controllare quanto spazio occupa realmente il nostro file: basterà dare un'occhiata alla directory.

Come si può notare dalla sezione riservata al controllo degli errori, in questa fase occorre tener presente che, dopo la preestensione, il Dos segnala l'errore 50 (*Record Not Present*), da ignorare senza problemi.

Nel programma viene anche effettuato un controllo se il file esiste già, cercando di aprire il file indice associato al file relativo.

Questo per evitare che, se presente, possa andare perduto qualche dato: riscrivendo infatti qualcosa in un record già occupato, se ne perde il contenuto precedente.

Una volta creato l'archivio, si può (o meglio, *si deve*) aprirlo tramite l'opzione 2.

Del successivo menu, di immediata comprensione, solo un paio di punti meritano qualche precisazione; per il resto il listato è più che sufficiente.

Soffermiamoci su questi.

ALLA RICERCA DEL DATO PERDUTO

Come si è detto, per estrarre un dato dal file basta segnalare al sistema, con opportuna sintassi, il numero del record, ed impartire un comando *Input#* o *Get#*.

Apparentemente è tutto semplice, ma sorgono due problemi:

1) Come selezionare il solo campo che interessa all'interno del record.

2) Dovendo ricercare il numero di telefono del Sig. Rossi (un John Smith nostrano), come identificare il numero di record in cui è memorizzato.

Bene, ecco pronta un'altra sfilza di *scoop*, almeno per i non addetti ai lavori.

Per rispondere al primo quesito è necessario un un ulteriore esame della struttura di un record.

STRUTTURA LOGICA

FILE	record 1	record 2	record ...	
RECORD	campo 1	campo 2	campo...	
CAMPO	byte 1	byte 2	byte 3	byte...

STRUTTURA FISICA

cognome	nome	citta'	cognome	nome	citta'	...

RECORD DI 16 BYTE CON 2 CAMPI

con campi di lunghezza variabile

campo 1						campo 2						vuoto			
R	O	S	S	I	*	M	A	R	I	O	*	/	/	/	/

con campi di lunghezza fissa

campo 1								campo 2							
R	O	S	S	I				M	A	R	I	O			*

spazi

* = Ritorno Carrello

FILE ARCHIVIO (CASUALE)

	record1	record2	record3
campo1	BIANCHI	ESPOSITO	ROSSI
campo2	Paolo	Gennaro	Mario
campo3	Torino	Napoli	Milano

FILE INDICE (SEQUENZIALE)

chiave	BIANCHI	ESPOSITO	ROSSI
posiz.	1	2	3

chiave = campo1

CAMBIARE IL CAMBIABILE

Il programma di queste pagine, pur abbastanza flessibile, si presta ad essere migliorato in più di una sua parte, purchè se ne sia prima ben capita la struttura.

In esso, infatti, viene privilegiato l'aspetto didattico, talvolta anche a scapito della velocità.

Un esempio banale: per eccesso di prudenza ogni settore operativo apre e poi chiude i files, mantenendoli aperti solo lo stretto necessario.

Questo per evitare che, magari per distrazione, si tolga per esempio il dischetto dal drive nel momento meno adatto.

Se ci si sente abbastanza sicuri delle proprie capacità, si può invece aprire il file relativo nelle prime righe del programma, per chiuderlo solo a conclusione dei lavori: rischioso, ma produttivo in termini di velocità.

Un'altra modifica, o meglio aggiunta, di facile implementazione, riguarda l'eventuale cancellazione di record già memorizzati.

In pratica basterà aggiungere una opzione di scrittura che scelga il numero di record da immettere.

Decidendo di scrivere qualcosa in un record già esistente, se ne cancellerà il contenuto precedente, e nulla vieta che, ad esempio, si memorizzi una serie di trattini (o il carattere che preferite) al posto di "veri" campi.

Un'altra chance, può riguardare la Lista Selettiva.

Questo è forse il punto più debole dell'archivio, in rapporto alla velocità.

Dal momento che è sfruttata la tecnica di archiviazione con campi di lunghezza variabile, in questa fase si è obbligati ad una lettura quasi sequenziale (vedi cicli *For ... Next* del listato).

L'unico modo per sveltire le operazioni consiste nel modificare la struttura generale dell'archivio, basandola sui campi a lunghezza fissa, e, perchè no, anche muniti di carattere separatore (Ritorno Carrello).

Si badi, però, che parallelamente si avrà un leggero rallentamento della fase di input da tastiera per le opzioni di lettura e scrittura.

Se si desidera paragonare il differente comportamento nelle due situazioni, tutto quello che si dovrà fare è aggiungere (o sovrapporre) le seguenti righe al programma pubblicato, valide solo per la versione in basic 7.0 (del C/128):

```
377 if l=ln(1) then 380
378 nm$ = nm$ + chr$(32): l=l+1: goto 377
487 if len (tr$(x)) = ln(x) then 490
488 tr$(x) = tr$(x) + chr$(32): l=l+1: goto 487
595 for x=1 to sl-1: b=b + ln (x) + 1: next
600 b=b+1: scnclr: dopen#5, (r$): gosub 625
605 for x=1 to mx: record#5, x, b
607 input#5, a$: record#5, x, b
```

Apportando tali modifiche, non si dimentichi che i files creati con un sistema non sono compatibili con l'altro.

I campi che contiene, infatti, possono essere organizzati in maniera diversa, come rilevabile in figura 2.

Si può scegliere, per esempio, di immagazzinarli (in fase di scrittura) secondo lunghezze variabili, purchè il totale dei caratteri non superi la dimensione stabilita per il record (nel nostro programma = 47).

In questo caso è indispensabile che le stringhe associate ai singoli campi contengano, come ultimo carattere, un *Chr$ (13)*, ovvero un *Return* (da conteggiare nel calcolo della dimensione del record), tenendo sempre presente che, dopo un *Print#*, viene comunque aggiunto, automaticamente, un *Chr$ (13)*.

Nel primo esempio in figura, dunque, per leggere i due campi occorrerà posizionare il puntatore sul primo byte, quindi effettuare due letture, per esempio con: *Input*#5, a$, b$

In alternativa è possibile memorizzare i campi con lunghezza fissa, riempiendo con spazi gli eventuali caratteri eccedenti (secondo esempio in figura 2).

Seguendo tale procedura non è strettamente necessario inserire caratteri di separazione come il *Chr$ (13)*, in quanto, conoscendo a priori l'esatta lunghezzza del campo, si può posizionare il puntatore all'inizio del dato che interessa, per poi leggerlo carattere per carattere con *Get#*.

O ancora (e più velocemente), con un singolo *Input#* si può assegnare ad un'unica variabile l'intero record, per poi suddividerlo da programma adoperando le funzioni legate alle stringhe (*Left$*... ecc.).

Nel listato di queste pagine viene adottato il primo metodo (lunghezza variabile), che può comunque essere facilmente modificato (vedi riquadro).

Infine non resta che esaminare la fase di ricerca del numero di record, peraltro applicata nel più semplice dei modi.

La pratica maggiormente in uso, anche nei Database più potenti, si basa sulla creazione di un cosiddetto *File Indice*, riferito ad uno dei campi del record, detto *campo chiave*.

Osservando la figura 3, tutto sarà più chiaro.

In pratica viene creato un file sequenziale *di appoggio* a quello relativo.

In esso viene memorizzato solo il contenuto di un campo, nel nostro caso quello *Cognome*, nonchè, come primo dato, il numero dei record già immessi nel file principale.

Quando, da menu, si apre il file relativo con l'opzione 2, viene anche caricato in memoria quello sequenziale, tramite un ciclo *For ... Next* il cui indice rappresenta il numero di record associato al campo chiave.

Ovviamente, in fase di scrittura, per ogni nuova archiviazione viene aggiornato un vettore contenente i campi chiave, nonchè un contatore sul numero totale dei record.

E' questo il motivo per cui si rende necessaria, come ultima operazione, una corretta chiusura dell'archivio, che deve comprendere l'aggiornamento del file indice.

Senza questa indispensabile "manovra", gli ultimi dati immessi non verrebbero più riconosciuti nella successiva sessione di lavoro.

Da notare che il C/128, grazie al comando *Trap*, consente di prevenire ogni eventualità di arresto del programma, senza prima avere chiuso l'archivio.

Con il C/64, invece, anche solo per la pressione del tasto Run / Stop, si rischia di perdere i dati immessi.

In questa eventualità, basterà comunque impartire *Goto 335* in modo diretto.

Un'ultima considerazione.

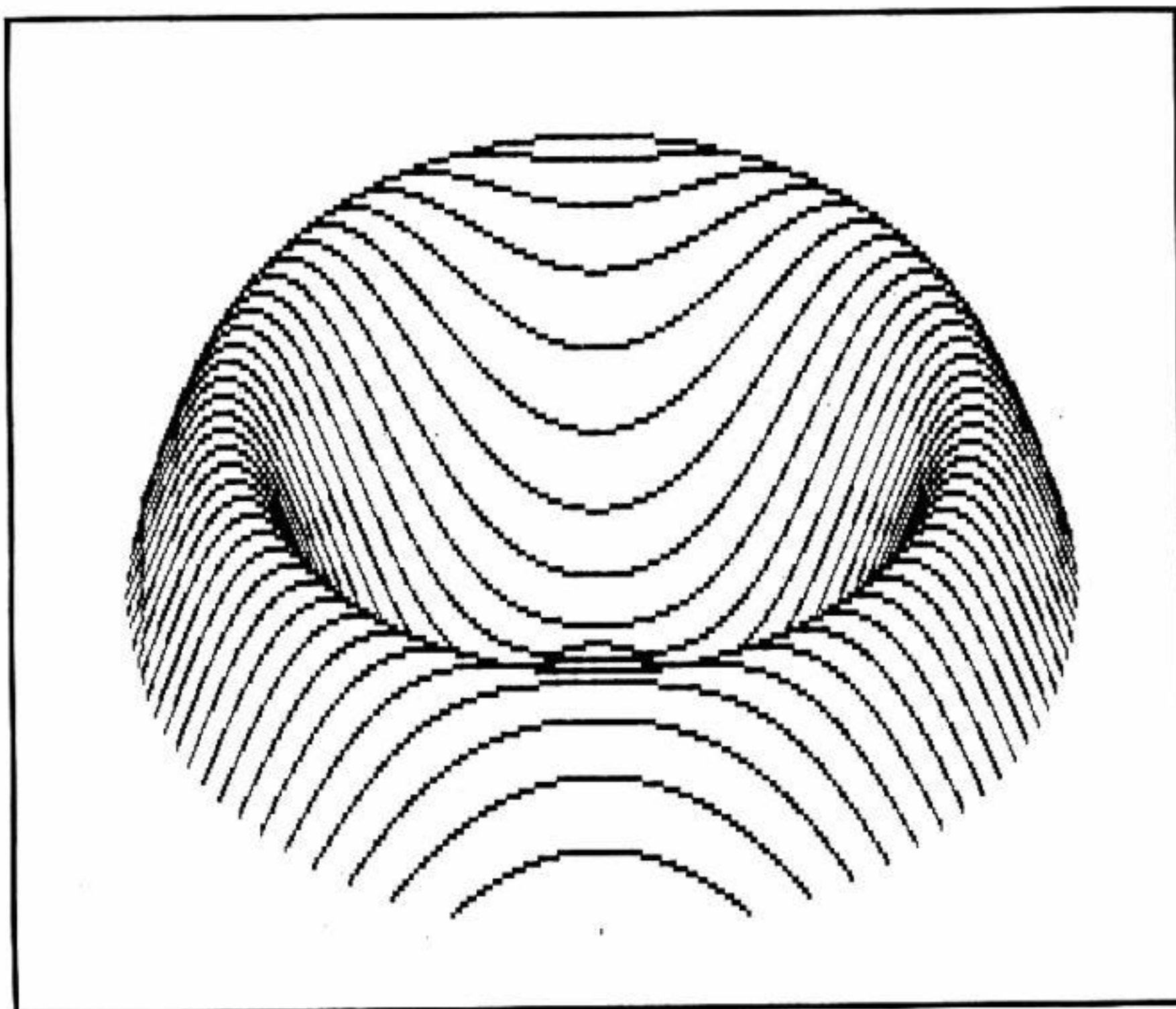
La fase di ricerca, così impostata, pur efficace e di semplice applicazione, non è tuttavia il massimo, soprattutto in presenza di grandi quantità di dati.

Esistono, a tal proposito, algoritmi di particolare velocità ed efficienza, che potrebbero rendere di livello professionale il programma qui proposto.

```
100 REM==================================
105 REM GESTIONE REL FILES IN BASIC 2.0
110 REM==================================
115 REM=== ASSEGNAZIONI MODIFICABILI ===
120 R$="RUBRICA":K$=R$+".KEY"
125 R=100:D=47:NC=4:DIM CP$(NC),LN(NC)
130 CP$(1)="COGNOME ":LN(1)=16
135 CP$(2)="NOME     ":LN(2)=16
140 CP$(3)="PREFISSO":LN(3)=4
145 CP$(4)="TELEFONO":LN(4)=7
150 REM==================================
155 DIM KE$(R),TR$(NC):OPEN15,8,15
160 FL$="*":S$=CHR$(147)
165 PRINTS$:PRINT"1 - CREAZIONE EX NOVO"
170 PRINT"2 - APERTURA ARCHIVIO"
175 IFAP=0THEN220
180 IFPP=1THEN190
185 PRINT"3 - IMMISSIONE DATI"
190 PRINT"4 - LETTURA DATI"
195 PRINT"5 - LISTA SELETTIVA"
200 PRINT"6 - CHIUSURA ARCHIVIO"
205 IFAP=0THEN220
210 PRINT:PRINTCHR$(18);
215 PRINT"RECORD ARCHIVIATI ="MX
220 GETA$:A=VAL(A$):IFA$=""THEN220
225 IFA<1ORA>6THEN220
230 ONAGOTO 240,295,460,365,580,335
235 REM======= CREAZIONE EX NOVO =======
240 OPEN3,8,3,K$:GOSUB645:IFE=62THEN255
245 PRINTS$:PRINT"L'ARCHIVIO ESISTE!!"
250 CLOSE3:PRINT:GOSUB700:PRINTS$:GOTO170
255 CLOSE3:PRINTS$:PRINT"ATTENDI..."
260 OPEN3,8,3,K$+",S,W":PRINT#3,FL$:CLOSE3
265 REM--------- PREESTENSIONE ---------
270 OPEN5,8,5,R$+",L,"+CHR$(D):GOSUB655
275 RR=R:GOSUB720:GOSUB655
280 PRINT#5,CHR$(255):GOSUB655:CLOSE5
285 PRINTS$:GOTO170
290 REM====== APERTURA ARCHIVIO ========
295 OPEN3,8,3,K$:GOSUB645:IFE<>62THEN310
300 PRINTS$:PRINT"ARCHIVIO NON PRESENTE"
305 PRINT:GOSUB700:RUN
310 PRINTS$:INPUT#3,MX$:GOSUB655
315 IFMX$=FL$THENMX=0:CLOSE3:AP=1:GOTO180
320 MX=VAL(MX$):FORX=1TOMX:INPUT#3,KE$(X)
325 NEXT:CLOSE3:AP=1:GOTO180
330 REM====== CHIUSURA ARCHIVIO ========
335 IFCB=0THENPRINTS$:GOTO685
340 PRINTS$:PRINT"ATTENDI..."
345 PRINT#15,"S:"+K$:OPEN3,8,3,K$+",S,W"
350 GOSUB655:PRINT#3,STR$(MX):FORX=1TOMX
355 PRINT#3,KE$(X):NEXT:GOTO685
360 REM======== LETTURA DATI ===========
365 PRINTS$:IFAP=0THENGOSUB695:RUN
370 IFMX=0THENGOTO180
375 NM$="":PRINTCP$(1);:INPUTNM$
380 IFNM$=""THENPRINTS$:GOTO180
385 L=LEN(NM$):IFL>LN(1)THEN375
390 FORX=1TOMX
395 IFKE$(X)<>NM$THEN450
400 PRINTS$:PRINT"RECORD N."X:PRINT:PRINT
405 OPEN5,8,5,R$+",L,"+CHR$(D)
410 GOSUB655:RR=X:GOSUB720
415 GOSUB655:FORY=1TONC:INPUT#5,TR$(Y)
420 PRINTCHR$(18)CP$(Y)CHR$(146),
425 PRINTTR$(Y):NEXTY:CLOSE5
430 PRINT:PRINT"CONTINUO RICERCA? (S/N)"
435 GETA$:IFA$=""THEN435
440 IFA$<>"S"ANDA$<>"N"THEN435
445 IFA$="N"THENPRINTS$:GOTO180
450 NEXTX:PRINTS$:GOTO180
455 REM==== INPUT DATI PER SCRITTURA ==
460 PRINTS$:IFAP=0THENGOSUB695:RUN
465 IFMX+1=<RTHEN485
470 PRINTS$:PRINT"ARCHIVIO PIENO!":PRINT
475 GOSUB700:PP=1:PRINTS$:GOTO190
480 REM----------------------------------
485 PRINTS$:PRINT"RECORD N."MX+1:PRINT
490 PRINT:FORX=1TONC
495 TR$(X)="":PRINTCP$(X);:INPUTTR$(X)
500 IFTR$(X)=""THENPRINTS$:GOTO180
505 IFLEN(TR$(X))>LN(X)THEN495
510 IFX=NCTHEN520
515 TR$(X)=TR$(X)+CHR$(13)
520 AR$=AR$+TR$(X):NEXT
525 PRINT:PRINT"MODIFICHE? (S/N)"
530 GETA$:IFA$=""THEN530
535 IFA$<>"S"ANDA$<>"N"THEN530
540 IFA$="S"THENAR$="":GOTO485
545 REM---------- SCRITTURA -------------
550 MX=MX+1:CB=1
555 KE$(MX)=LEFT$(TR$(1),LEN(TR$(1))-1)
560 OPEN5,8,5,R$+",L,"+CHR$(D):GOSUB655
565 RR=MX:GOSUB720:GOSUB655:PRINT#5,AR$
570 GOSUB655:AR$="":CLOSE5:PRINTS$:GOTO180
575 REM====== LISTA SELETTIVA =========
580 PRINTS$:IFAP=0THENGOSUB695:RUN
585 IFMX=0THEN180
590 FORX=1TONC:PRINT"CAMPO"X"= "CP$(X)
595 NEXT:PRINT:PRINT:SL=0
600 PRINT"NUMERO DEL CAMPO (E RETURN)";
605 INPUTSL:IFSL>NCTHENSL=0:GOTO600
610 PRINTS$:IFSL=0THEN180
615 OPEN5,8,5,R$+",L,"+CHR$(D):GOSUB655
620 FORX=1TOMX:RR=X:GOSUB720:FORY=1TOSL
625 INPUT#5,A$:NEXT
630 PRINTKE$(X)TAB(17)A$:NEXT:CLOSE5
635 PRINT:GOSUB700:B=0:PRINTS$:GOTO180
640 REM===== CONTROLLO ERRORI ==========
645 INPUT#15,E,E$:IFE=0ORE=62THENRETURN
650 GOTO660
655 INPUT#15,E,E$:IFE=0ORE=50THENRETURN
660 PRINTS$:PRINT"ERRORE!":PRINTE$
665 PRINT:PRINT"CHIUDO IL FILE? (S/N)"
670 GETA$:IFA$=""THEN670
675 IFA$<>"S"ANDA$<>"N"THEN670
680 IFA$="S"THEN335
685 CLOSE3:CLOSE5:CLOSE15:PRINTS$:END
690 REM----------------------------------
695 PRINT"ARCHIVIO NON APERTO!":PRINT
700 PRINT:PRINT"PREMI UN TASTO"
705 GETA$:IFA$=""THEN705
710 RETURN
715 REM==== POSIZIONA PUNTATORE=========
720 HI=INT(RR/256):LOW=RR-HI*256
725 PRINT#15,"P"+CHR$(5)+CHR$(LO)+CHR$(HI)+CHR$(1)
730 RETURN
```

```
100 REM================================
105 REM GESTIONE REL FILES IN BASIC 7.0
110 REM================================
115 TRAP 630
120 REM--- ASSEGNAZIONI MODIFICABILI ---
125 R$="RUBRICA":K$=R$+".KEY"
130 R=100:D=47:NC=4:DIM CP$(NC),LN(NC)
135 CP$(1)="COGNOME  ":LN(1)=16
140 CP$(2)="NOME     ":LN(2)=16
145 CP$(3)="PREFISSO":LN(3)=4
150 CP$(4)="TELEFONO":LN(4)=7
155 REM--------------------------------
160 FL$="*":DIM KE$(R),TR$(NC):SCNCLR
165 PRINT"1 - CREAZIONE EX NOVO"
170 PRINT"2 - APERTURA ARCHIVIO"
175 IFAP=0THEN215
180 IFPP=1THEN190
185 PRINT"3 - IMMISSIONE DATI"
190 PRINT"4 - LETTURA DATI"
195 PRINT"5 - LISTA SELETTIVA"
200 PRINT"6 - CHIUSURA ARCHIVIO"
205 IFAP=1THENBEGIN:PRINT:PRINTCHR$(18);
210 BEND:PRINT"RECORD ARCHIVIATI  ="MX
215 GETKEYA$:A=VAL(A$):IFA<1ORA>6THEN215
220 ONAGOTO 230,285,440,355,560,325
225 REM------- CREAZIONE EX NOVO -------
230 DOPEN#3,(K$):IFDS=62THEN250
235 GOSUB625:SCNCLR:DIRECTORY(R$+FL$)
240 PRINT:PRINT"L'ARCHIVIO ESISTE!!"
245 DCLOSE:PRINT:GOSUB670:SCNCLR:GOTO170
250 DCLOSE:SCNCLR:PRINT"ATTENDI..."
255 DOPEN#3,(K$),W:PRINT#3,FL$:DCLOSE
260 REM--------- PREESTENSIONE ---------
265 DOPEN#5,(R$),L(D):GOSUB625:RECORD#5,R
270 GOSUB625:PRINT#5,CHR$(255):GOSUB625
275 RECORD#5,R:DCLOSE:SCNCLR:GOTO170
280 REM------ APERTURA ARCHIVIO --------
285 DOPEN#3,(K$):IFDS=62THENBEGIN
290 SCNCLR:PRINT"ARCHIVIO NON PRESENTE"
295 BEND:PRINT:GOSUB670:RUN
300 SCNCLR:GOSUB625:INPUT#3,MX$:GOSUB625
305 IFMX$=FL$THENMX=0:DCLOSE:AP=1:GOTO180
310 MX=VAL(MX$):FORX=1TOMX:INPUT#3,KE$(X)
315 NEXT:DCLOSE:AP=1:GOTO180
320 REM------ CHIUSURA ARCHIVIO --------
325 IFCB=0THENSCNCLR:GOTO655
330 SCNCLR:PRINT"ATTENDI..."
335 SCRATCH(K$):DOPEN#3,(K$),W:GOSUB625
340 PRINT#3,STR$(MX):FORX=1TOMX
345 PRINT#3,KE$(X):NEXT:GOTO655
350 REM-------- LETTURA DATI -----------
355 SCNCLR:IFAP=0THENGOSUB665:RUN
360 IFMX=0THENGOTO180
365 NM$="":PRINTCP$(1);:INPUTNM$
370 IFNM$=""THENSCNCLR:GOTO180
375 L=LEN(NM$):IFL>LN(1)THEN365
380 FORX=1TOMX
385 IFKE$(X)=NM$THENBEGIN:SCNCLR
390 PRINT"RECORD N."X:PRINT:PRINT
395 DOPEN#5,(R$):GOSUB625:RECORD#5,(X)
400 GOSUB625:FORY=1TONC:INPUT#5,TR$(Y)
405 PRINTCHR$(18)CP$(Y)CHR$(146),
410 PRINTTR$(Y):NEXTY:RECORD#5,(X):DCLOSE
415 PRINT:PRINT"CONTINUO RICERCA? (S/N)"
420 GETKEYA$:IFA$<>"S"ANDA$<>"N"THEN420
425 BEND:IFA$="N"THENSCNCLR:GOTO180
430 NEXTX:SCNCLR:GOTO180
435 REM==== INPUT DATI PER SCRITTURA --
440 SCNCLR:IFAP=0THENGOSUB665:RUN
445 IFMX+1>RTHENBEGIN:SCNCLR
450 PRINT"ARCHIVIO PIENO!":PRINT:PRINT
455 BEND:GOSUB670:PP=1:SCNCLR:GOTO190
460 REM--------------------------------
465 SCNCLR:PRINT"RECORD N."MX+1:PRINT
470 PRINT:FORX=1TONC
475 TR$(X)="":PRINTCP$(X);:INPUTTR$(X)
480 IFTR$(X)=""THENSCNCLR:GOTO180
485 IFLEN(TR$(X))>LN(X)THEN475
490 IFX=NCTHEN500
495 TR$(X)=TR$(X)+CHR$(13)
500 AR$=AR$+TR$(X):NEXT
505 PRINT:PRINT"MODIFICHE? (S/N)"
510 GETKEYA$:IFA$<>"S"ANDA$<>"N"THEN510
515 IFA$="S"THENAR$="":GOTO465
520 REM--------- SCRITTURA ------------
525 MX=MX+1:CB=1
530 KE$(MX)=LEFT$(TR$(1),LEN(TR$(1))-1)
535 DOPEN#5,(R$):GOSUB625
540 RECORD#5,(MX):GOSUB625:PRINT#5,AR$
545 GOSUB625:RECORD#5,(MX):GOSUB625
550 AR$="":DCLOSE:SCNCLR:GOTO180
555 REM------ LISTA SELETTIVA ----------
560 SCNCLR:IFAP=0THENGOSUB665:RUN
565 IFMX=0THEN180
570 FORX=1TONC:PRINT"CAMPO"X"= "CP$(X)
575 NEXT:PRINT:PRINT:SL=0
580 PRINT"NUMERO DEL CAMPO (E RETURN)";
585 INPUTSL:IFSL>NCTHENSCNCLR:GOTO580
590 IFSL=0THENSCNCLR:GOTO180
595 SCNCLR:DOPEN#5,(R$):GOSUB625
600 FORX=1TOMX:RECORD#5,X:FORY=1TOSL
605 INPUT#5,A$:NEXT:RECORD#5,X
610 PRINTKE$(X)TAB(17)A$:NEXT:DCLOSE
615 PRINT:GOSUB670:B=0:SCNCLR:GOTO180
620 REM----- CONTROLLO ERRORI ----------
625 IFDS=0ORDS=50THENRETURN
630 SCNCLR:PRINT"ERRORE!":PRINT"DOS= "DS$
635 IFER>0THENPRINT"BAS= ";ERR$(ER)
640 PRINT:PRINT"CHIUDO IL FILE? (S/N)"
645 GETKEYA$:IFA$<>"S"ANDA$<>"N"THEN645
650 IFA$="S"THEN325
655 DCLOSE:SCNCLR:PRINTDS$:HELP:END
660 REM--------------------------------
665 PRINT"ARCHIVIO NON APERTO!":PRINT
670 PRINT"PREMI UN TASTO":GETKEYA$:RETURN
```

ACCETTA LA SFIDA!

Un dischetto, che puoi trovare in edicola proprio in questi giorni, propone una sfida adatta a veri campioni

Con la pausa estiva, si sa, non è possibile, nè auspicabile, cimentarsi in giochi complicati o cervellotici.

Alcuni patiti del computer, tuttavia, aspettano con ansia proprio il periodo delle ferie per buttarsi a capofitto sulla tastiera, ritenendola più rilassante di una passeggiata all'interno del tradizionale carnaio di ferragosto.

Bene, non possiamo dimenticare che esiste anche questa categoria di persone ed, anzi, non ci vergognamo ad ammettere che anche noi della Redazione aspettiamo le ferie per dedicarci a lavori informatici molto personali che, con molta probabilità, non vedranno mai la luce...

Capita quindi a proposito la decisione di divulgare, sul dischetto "Commodore Club" presente in tutte le edicole d'Italia, un particolare programma in grado di disegnare funzioni matematiche tridimensionali.

Come è noto, un qualsiasi grafico deriva dallo sviluppo grafico di una funzione matematica che, spesso, richiede l'elaborazione di tre variabili.

Il risultato consiste nelle meravigliose figure che spesso vediamo apparire in moltissimi programmi, non sempre di spirito matematico.

Una certa funzione, inoltre, presenta grafici profondamente diversi tra loro a seconda dell'impostazione di alcuni parametri; basta scrivere un 10, invece di un 11, per veder apparire un grafico del tutto diverso dal precedente.

IL PROGRAMMA

Il programma presente sul dischetto di cui stiamo parlando è in grado di visualizzare quattro diverse funzioni matematiche, per ciascuna delle quali è possibile impostare quattro parametri diversi. Tenendo conto che il range dei parametri è pressochè infinito, è facile immaginare la quantità di figure che si possono riprodurre su video e su carta, dal momento che il listato dispone di quest'utile funzione.

DOPPIA SFIDA

In queste pagine vedete pubblicati alcuni output ottenuti con il listato di cui sopra. Una prima sfida consiste nell'individuare il tipo di funzione da selezionare ed i parametri da assegnare per ottenere almeno uno dei grafici qui riportati. Dal momento che l'impresa è piuttosto ardua, riterremo validi la funzione ed i parametri che siano in grado di riprodurre una schermata il più possibile simile ad una di quelle qui riprodotte.

La seconda sfida, invece, terrà occupati i lettori che, non volendo utilizzare il di-

schetto, dedicheranno il loro tempo per individuare la funzione ed i parametri in grado di tracciare uno dei grafici pubblicati. In questo caso, però, dovranno far pervenire in Redazione non solo la funzione, ma anche il programma (in qualsiasi linguaggio: Basic 64, Gw-Basic, T. Pascal, Simon's Basic, Basic 7.0, Basic 3.5 e così via) che sia in grado di far apparire su schermo quanto richiesto.

COME ACCETTARE LA SFIDA

Se intendete usare il programma presente su dischetto, è sufficiente scrivere su un semplice foglio di carta (da inviare in busta chiusa, affrancata secondo le vigenti tariffe postali) il tipo di funzione da selezionare ed il valore dei quattro parametri da impostare.

I lettori che, invece, vogliono sperimentare per proprio conto, dovranno inviare (su disco) il programma in grado di far apparire il grafico.

Ai primi 100 lettori che faranno pervenire la soluzione entro il 31 agosto 1989 verrà inviato un simpatico omaggio.

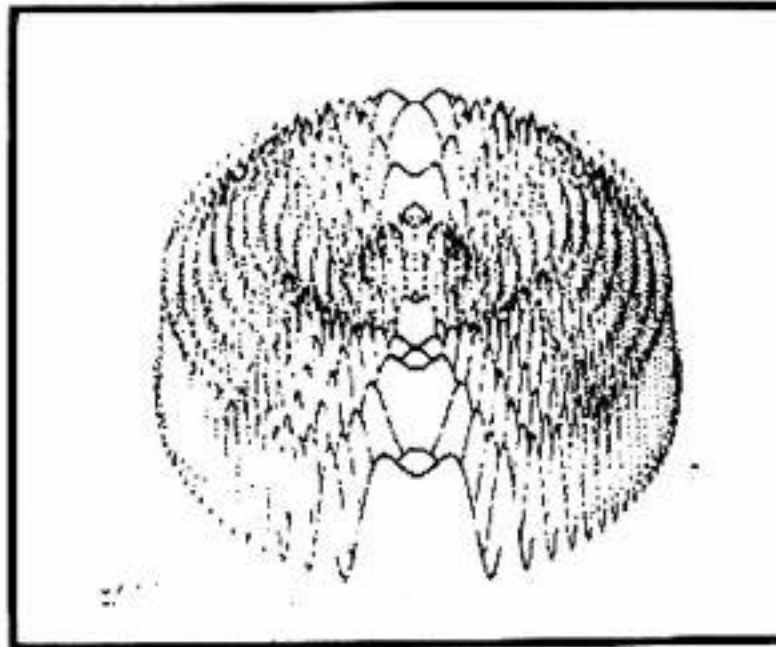
GLI ALTRI FILE

Mercenaries

Fama, gloria e ricchezza.

Così si presenta il futuro di fronte agli occhi dell'intrepido mercenario pronto a tutto pur di guadagnarsi un posto di rispettabile condottiero nell'esercito imperiale di Karr.

La via è molto semplice ed è aperta a tutti; si tratta semplicemente di arrivare nel porto interstellare di Starview e da lì, una volta adeguatamente equipaggiati, partire alla volta di una delle sedici basi nemiche poste ai confini del braccio galattico. Molti hanno tentato questa via ma soltanto pochi sono riusciti a percorrerla interamente fino a guadagnarsi un posto di elevato rango nell'esercito imperiale; l'imperatore Taurus I stesso pare provenga da una carriera simile anche se allora, quando la galassia era giovane, non esisteva nemmeno un nemico così formidabile come i Commox.

La tua avventura comincia nel porto spaziale di Starview dove dovrete darvi da fare per acquistare l'equipaggiamento necessario a portare a termine le vostre missioni; il denaro che possedete è il frutto di anni di sacrifici vostri e della vostra famiglia, per cui fate molta attenzione a spenderlo in modo appropriato, senza sperperare e tuttavia senza "stringere" troppo la cinghia; avrete modo di guadagnare maggiormente in seguito, una volta che porterete a termine una missione.

Andate ora! Fama, gloria e ricchezza vi aspettano in giro per l'universo; a voi la possibilità di riuscire o di fallire miseramente restando per sempre su una base aliena.

Ricordatevi che, una volta partiti, non si può più cambiare idea... l'avventura vi aspetta.

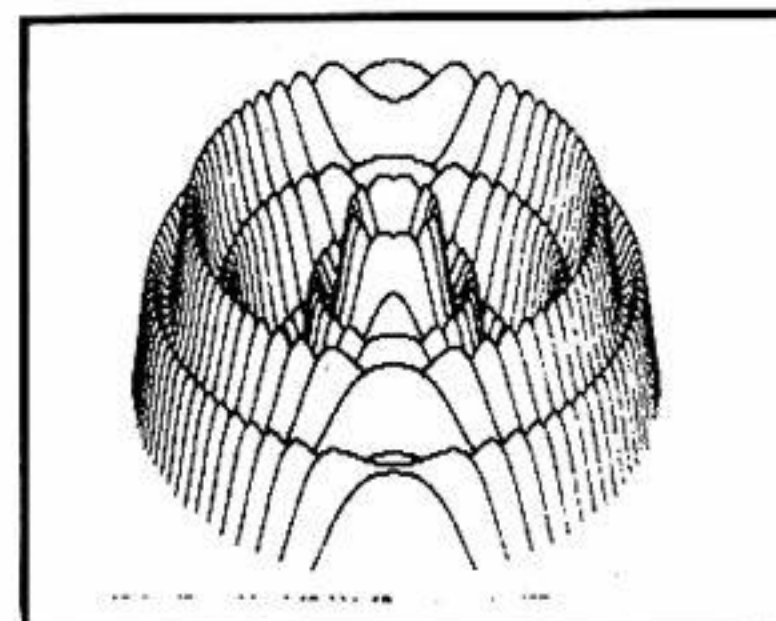
Out of deep

Cosa mai può spingere un animaletto simpatico come il magico pipistrello di Rauack nelle viscere della terra?

La curiosità, ovviamente! Il cunicolo era troppo invitante e la giornata si faceva sempre più afosa e torrida.. E poi c'era la luce; tanta luce, troppa per chi è abituato a girare di notte.

L'indecisione durò soltanto un attimo; il pipistrello si inoltrò quindi nelle oscurità della terra attratto dalla incredibile sensazione di fresco che lo portava sempre più in basso.

Improvvisamente si accorse di essere sceso troppo, molto più di quanto lui avesse mai fatto nel corso degli ultimi secoli; improvvisamente si trovò in una grande caverna che pullulava di mostri di ogni genere (orchetti, pesci d'aria amebe e così via).

L'istinto prese il sopravvento ed immediatamente fece dietro-front verso il cunicolo appena percorso ma, amara sorpresa, questo era bloccato da un enorme drago fiammeggiante che lo aspettava a fauci aperte; non era possibile passare di là.

Non restava altra via che la risalita di quella caverna immensa e così il pipistrello cominciò quella che sarebbe stata la sua più grande avventura: l'uscita dall'abisso.

Cambogia

Nebbia, pesante ed umida.

Una coltre impenetrabile avvolge l'accampamento militare intorpidito dal sonno della notte; la foresta silenziosa osserva impassibile i pesanti sforzi compiuti dalla compagnia di soldati intenti ad organizzare una nuova dura giornata di esplorazione.

Lontano dall'accampamento il tenente Mohani cerca disperatamente una via veloce per raggiungere l'alto comando delle forze di liberazione; il messaggio che trasporta è di vitale importanza per la riuscita dell'operazione bellica in corso che porrà fine in modo definitivo alle ostilità che devastano il paese.

Il vostro equipaggiamento comprende un mitragliatore leggero e 100 colpi che dovrete fare attenzione a non sprecare inutilmente; fortunatamente lungo il cammino troverete una serie di caricatori da 10 colpi che potrete utilizzare per difendervi dai continui attacchi del nemico che cercherà di ostacolarvi in tutti i modi possibili, cercando di utilizzare quanti più uomini e mezzi potrà.

Attenzione, per proseguire dovrete abbattere tutti i nemici che vi si pareranno davanti, altrimenti verrete ucciso e la vostra missione sarà fallita.

A!, quasi dimenticavamo: sullo stesso dischetto sono presenti, come al solito, tutti i programmi (formato C/64, C/16, C/128) pubblicati su "Commodore Computer Club".

L'INDIRIZZO

Il recapito cui inviare le vostre risposte è il seguente:

Commodore Computer Club
("Sfida grafica" estate '89)

c/o SYSTEMS Editoriale
Via Mosè, 18
20090 OPERA (Mi)

risposta al QUIZ

(Risposta ai quiz di pag. 31)

PRIMO LIVELLO

1* D sarà uguale a 23, mentre E varrà 35.

2* Basta dividere per due la variabile X, e controllare se il risultato è un valore intero tramite la funzione INT. In pratica:

```
5 INPUT X
10 IF X/2 = INT(X/2) THEN PRINT
"PARI"
```

3* Considerando che 320 e 200 indicano la dimensione (espressa in pixel) orizzontale e verticale della schermata, e che ad ogni pixel deve corrispondere un bit in memoria, avremo: 320*200 = 64000 bit. Poichè ogni locazione di memoria contiene 8 bit (= 1 byte), necessiteranno, in definitiva, 64000 / 8 = 8000 locazioni.

4* A = INT (RND(0) * 40 + 1)

5* Eccone un esempio...
```
10 OPEN 15, 8, 15: INPUT#15, A, A$
20 PRINTA; A$: CLOSE 15
```
...con il nome delle variabili che, ovviamente, può essere variato, ma con l'obbligo dei numeri di linea, in quanto non è possibile digitare la sequenza in modo diretto.

6* Perchè, quando l'interprete del Basic incontra una variabile con indice, ne garantisce automaticamente un dimensionamento di 11 elementi (numerati, però, da 0 a 10). Oltre tale numero, è necessaria una istruzione Dim. Nel secondo caso, quindi, manca il preventivo comando DIM A(20).

7* Con PRINT CHR$(34).

8* B$ = MID$(A$, 11, 8).

9* ROM = Read Only Memory (Memoria a sola lettura). RAM = Random Access Memory (Memoria ad accesso casuale).

10* Division by zero error in 20.

LIVELLO AVANZATO

11* Praticamente un numero infinito di volte, fino a che non si arresta volontariamente il programma.

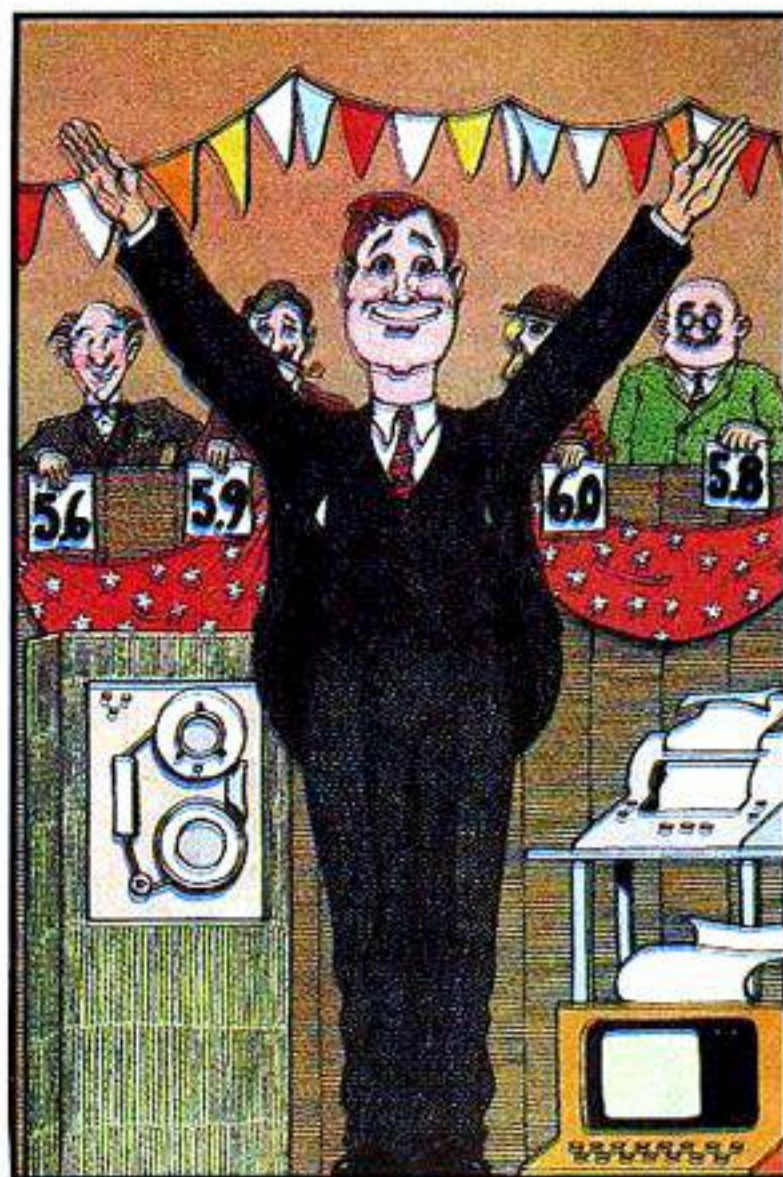
12* Sì, usandoli esattamente come i file sequenziali (SEQ), ma adoperando "P" al posto di "S" nella sintassi di apertura del file. Per esempio :
OPEN 5, 8, 5, "Nomefile, P, R"

13* Il numero massimo di caratteri consentiti per una stringa è 255, quindi l'errore sarà "String too long error in 20", mentre un Print X dopo l'arresto del programma fornirà il valore 256.

14* LOAD "$$", 8 (seguito dal consueto List).

15* L'errore sarà "Out of memory error in 10", ed apparirà dopo la visualizzazione del numero 24. Si badi, però, che l'effettivo numero di richiami possibili della stessa subroutine è 22, in quanto X viene incrementato prima dell'istruzione Gosub. Il Return, tra l'altro, non viene mai effettivamente eseguito.

TOP LEVEL

16* Una struttura composta da alcuni blocchi di un dischetto, che viene creata e utilizzata dal Dos quando si manipolano i files relativi (REL).

17* I primi due contengono l'indirizzo in memoria della riga successiva (Link), mentre la seconda coppia riporta il numero di riga. In entrambi i casi, i valori sono memorizzati nel formato basso / alto.

18* Token.

19* 53248 ($D000).

20* "Consultando" le locazioni di memoria 71 e 72, che puntano (in formato basso / alto) l'indirizzo in cui è memorizzato il valore della variabile. Nei due byte che precedono è rintracciabile il nome della stessa. Ecco un esempio sul modo corretto di operare :

```
10 X = 100
20 Y = X: REM Preserva X
30 X = PEEK(72) * 256 + PEEK(71)
40 PRINT "X E' ALLOCATA IN"; X-2
50 X = Y: REM Ripristina X
```

DOMANDA BONUS

* L'operazione 255 or X, quale che sia il valore di X, darà sempre 255 come risultato.

PUNTEGGI RAGGIUNTI

Meno di 3
No comment. Evidentemente hai iniziato da poco ad usare il computer, ma non disperare; vedrai che, impegnandoti un po', un giorno non lontano sarai in grado di rispondere a tutte queste domande.

Da 3 a 7
Sei già ad un buon punto, ma è solo l'inizio: manca solo un po' di perseveranza.

Da 8 a 12
Niente male, sai sfruttare un C/64 decisamente più a fondo della media degli utenti.

Da 13 a 18
Che cosa aspetti a passare al linguaggio macchina, o ad una fascia superiore di computer?

Super Bonus
Ti sei tradito, hacker maledetto!

UN AUTORUN... ANIMATO

Uno sprite compare, e si muove, durante il caricamento dei vostri programmi

di **Gianluca Venturini**

La tecnica più diffusa per dotare un programma di autorun consiste nel registrare, assieme (e prima) dei dati del programma Basic stesso (occupanti le locazioni da 2049 in poi), la zona di memoria precedente. Ci riferiamo, in particolare, a quella contenente i vettori di sistema, dei quali alcuni vengono caricati con valori modificati per acconsentire al nostro scopo.

L'idea è venuta dando un'occhiata al manualetto del C/64: si nota che in tale porzione di memoria sono contenuti, tra l'altro, anche i vettori di interrupt IRQ (788-789). Sorge quindi spontanea l'idea: perchè non alterare anche questi vettori per ottenere "qualcosa" mentre il programma si sta caricando?

Effettuando diverse prove si può constatare che, durante il caricamento, la routine di queste pagine viene (ovviamente!) chiamata, anche se non proprio con la consueta regolarità a causa delle operazioni di I/O in corso.

Un piccolo problema, felicemente risolto, era quello di trovare un'area di memoria in cui installare la routine per i nostri scopi. La zona doveva trovarsi prima dei vettori stessi, perchè, una volta caricati, la routine doveva essere già presente onde evitare un meschino inchiodamento del sistema (e del caricamento stesso).

Come già detto, comunque, abbiamo reperito qualche byte ancora libero a partire da 673.

Cominciò così il lavoro per ottenere il programma che vedete in queste pagine e che rappresenta uno spunto per personalizzare i vostri caicamenti.

Il tutto, se non l'avete ancora intuito, ottenendo UN SOLO FILE, pur se da caricare con la sintassi ",8,1".

IL PROGRAMMA

Il programma chiederà di inserire il disco contenente il programma Basic da dotare di questo speciale interrupt. Inseritelo, e digitate, quando richiesto, il nome del programma stesso. A questo punto "Autorun interrupt" creerà automaticamente su disco un file dal nome "AI/..." (vedi riga 1210) che, ad un suo successivo caricamento, visualizzerà uno sprite-omino che cammina.

Il file oggetto "AI/..." sara così composto (ripetiamo che sarà necessario caricarlo con il suffisso ",8 ,1"): prima, seconda routine di interrupt; vettori (dei quali quello di IRQ modificato per puntare alla prima routine di interrupt e quello di OUTPUT modificato per puntare alla routine di RUN automatico ed eventualmente alla routine di antireset); terza e quarta routine di interrupt; routine eventuale di antireset e/o routine di run; i dati dello sprite (quattro diverse posizioni) che occupano la zona da 1024 a 1279; diverse centinaia di spazi, un messaggio (variabile a piacere) e, finalmente, il vostro programma Basic.

Il programma vi chiede anche se desiderate dotare il file di antireset. Attenzione, perchè in questo caso il programma non solo non si potrà bloccare, ma non dovrà invadere la memoria da 32768 in poi.

LE ROUTINES L.M.

La prima routine L.M. ad essere eseguita durante il caricamento è quella che, nel disassemblato, è denominata interrupt/1. Essa pulisce lo schermo, stampa un messaggio, prepara lo sprite e colora la prima pagina di memoria video con il colore del fondo per "mascherare" i caratteri che ivi verranno caricati; questi, infatti, costituiscono i dati dello sprite. Infine viene dirottato il vettore interrupt verso la seconda routine.

La seconda routine aspetta che vengano caricati i dati dello sprite (il cui limite inferiore è segnalato da uno zero posto nella locazione 1279) e, quando ciò è avvenuto, salta alla terza routine.

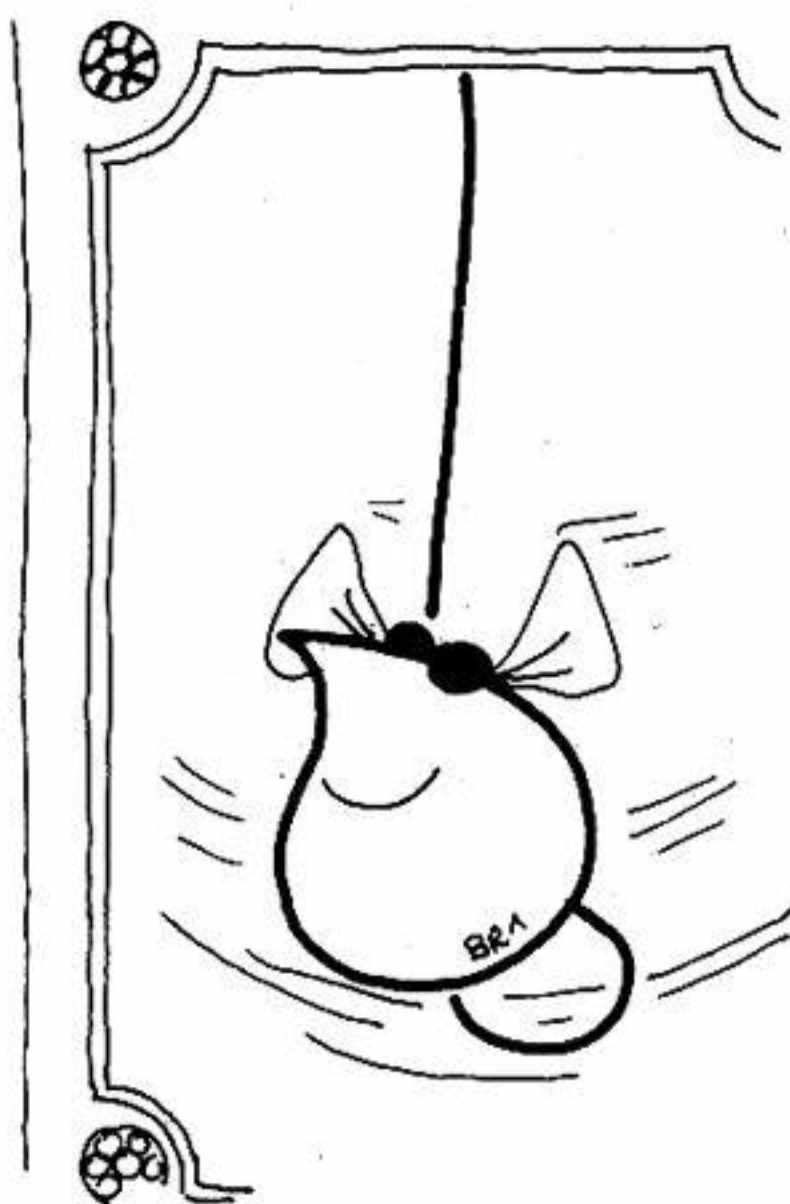
La terza routine visualizza lo sprite e dirotta il vettore interrupt verso la quarta routine, che si occupa di animare lo sprite; il movimento è, in realtà, fittizio in quanto non si fa altro che cambiare continuamente lo sprite scegliendolo tra quattro simili che rappresentano quattro attimi della "camminata" dell'omino.

Quest'ultima routine di interrupt (vi ricor-

diamo che, come si vede dai disssemblati, il normale ciclo di interrupt viene sempre ripreso al termne di ogni routine finora descritta) viene eseguita fino al termine del caricamento, momento in cui viene eseguito il ripristino, l'eventuale antireset e la routine di run, il cui indirizzo è puntato dai due byte del vettore di OUTPUT (806-807).

La routine "copia" serve invece al programma "Autorun interrupt" per trasferire, nel file oggetto AI/..., il programma sorgente, dopo i dati che verranno caricati da 673 a 2048.

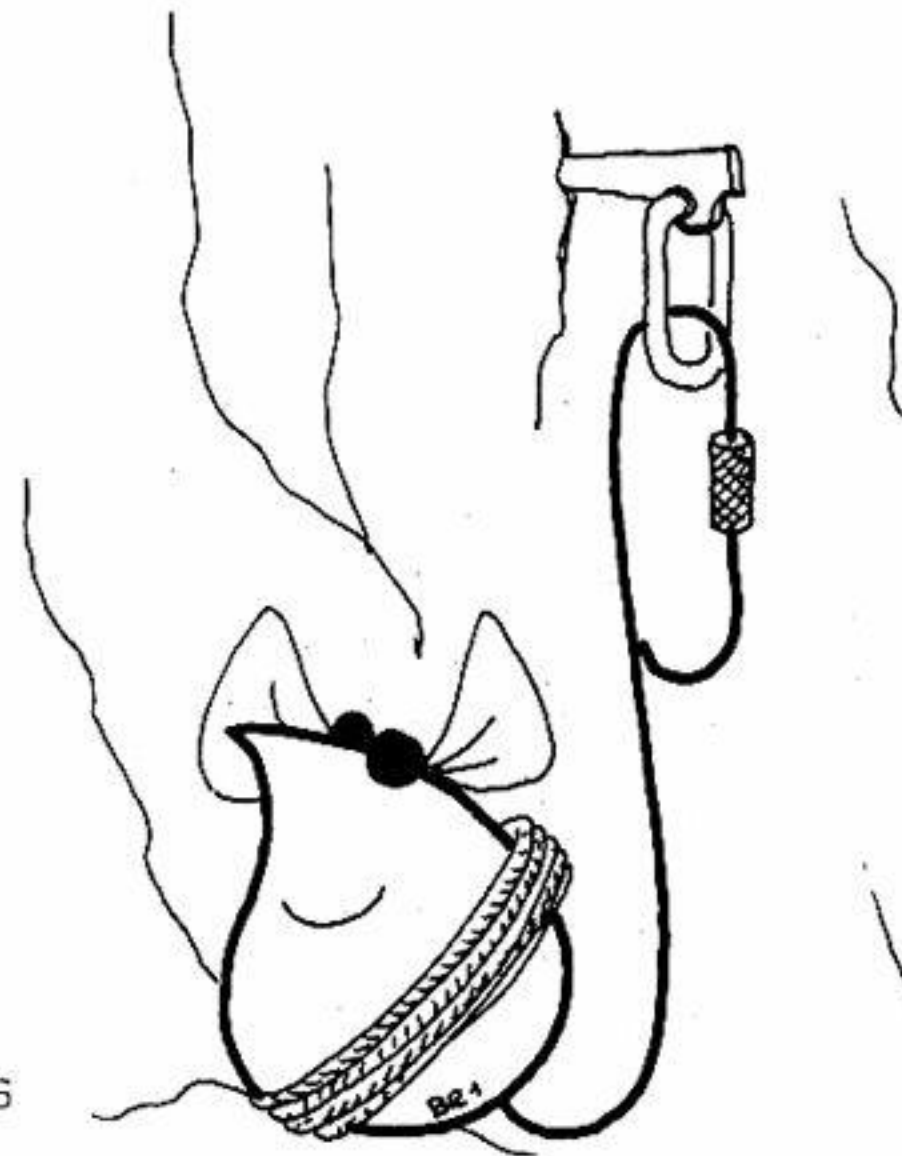
CONCLUSIONI

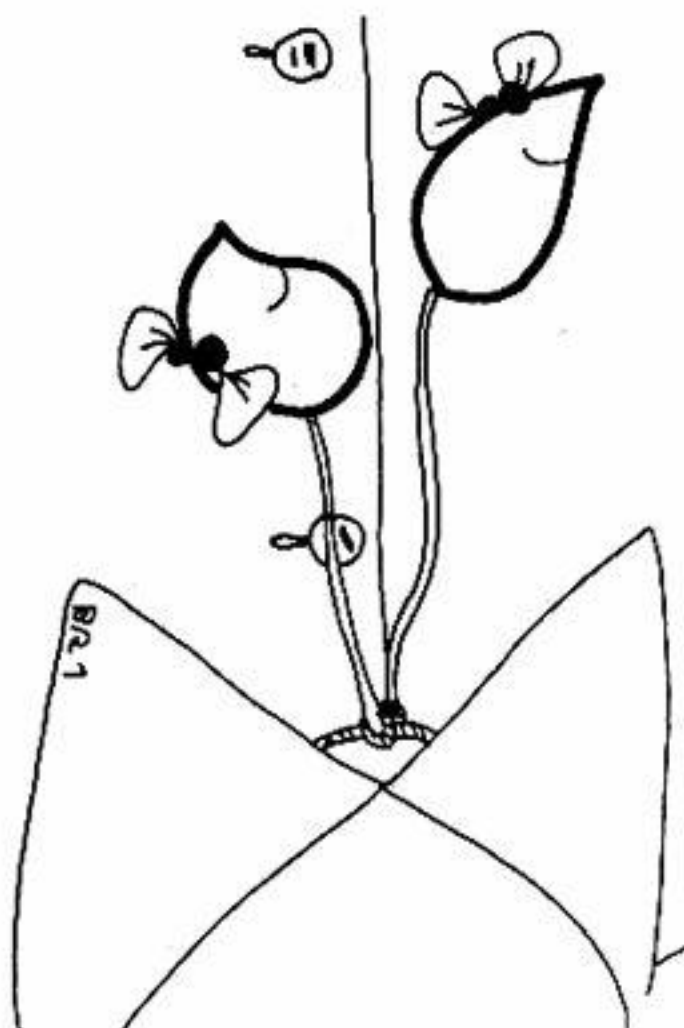
Se siete bravi, considerate l'esempio riportato come una delle possibili applicazioni di una nuova, simpatica tecnica di protezione.

Si potrebbe, tra l'altro, colorare lo schermo, stampare messaggi, visualizzare scenette, e tutto ciò che la vostra fantasia vi suggerisce.

Ricordatevi, però, che la frequenza di chiamata della routine di interrupt non è regolare, per cui dovrete fare i conti con questo inconveniente.

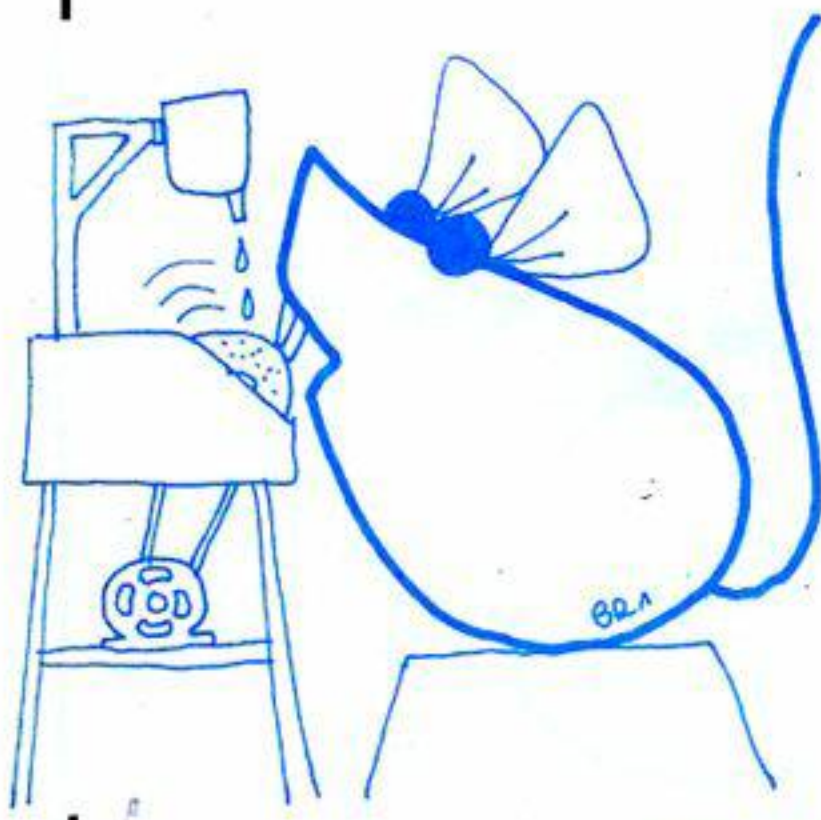
Modificando "pesantemente" la tecnica descritta, alcuni malfunzionamenti possono verificarsi perchè l'interrupt, richiamato automaticamente ogni 60mo di secondo, potrebbe essere attivato proprio nel momento successivo al caricamento del primo byte del vettore IRQ, quando, cioè, il secondo byte (dello stesso vettore) è ancora quello "vecchio". In questo caso l'interrupt verrebbe dirottato verso una routine inesistente o, quantomeno, priva di senso.

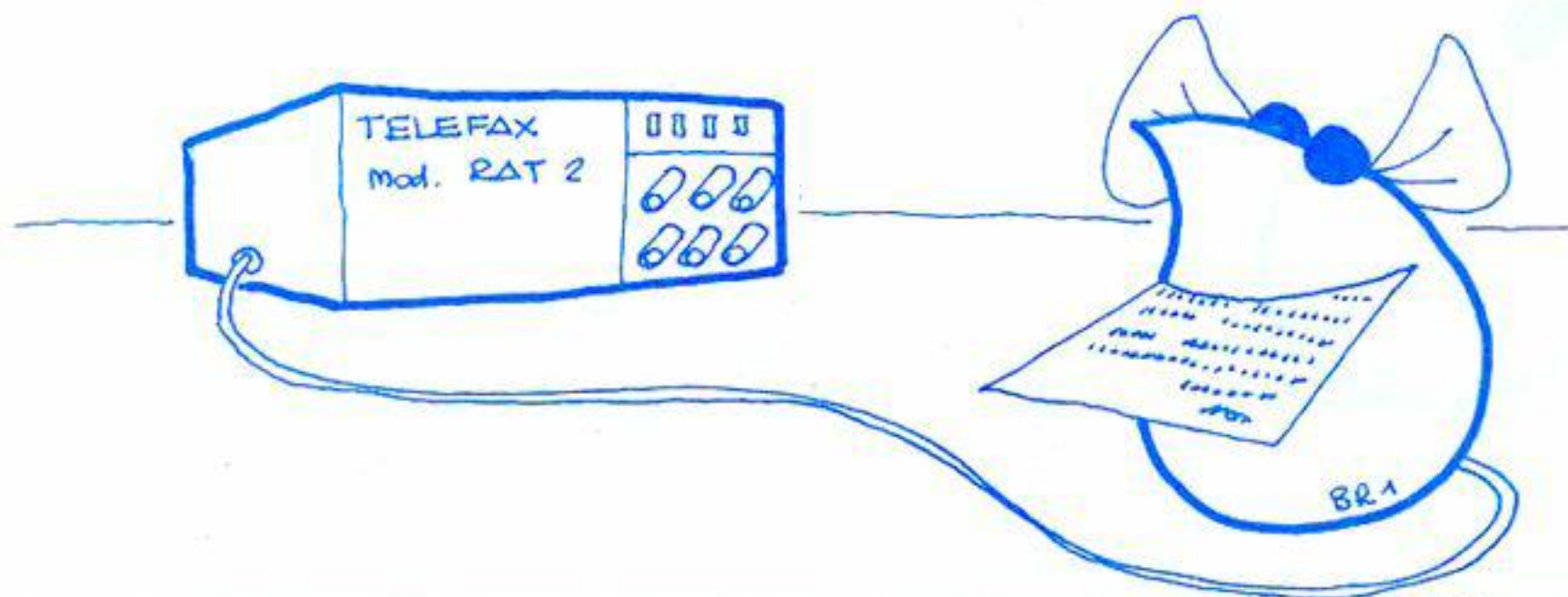
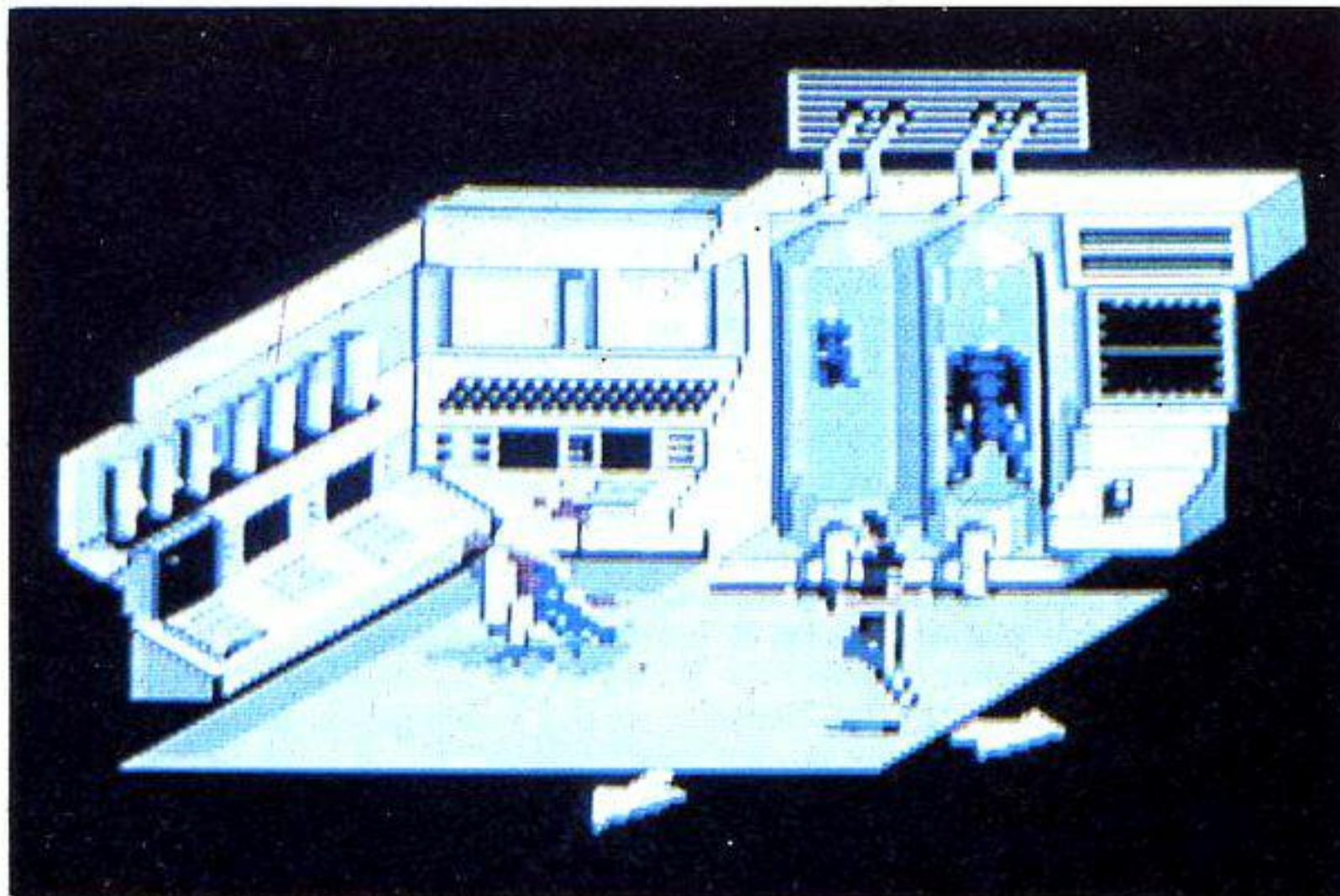
Non dimenticate, infine, che la presenza di molti sprite allineati in orizzontale può generare inchiodamenti durante la fase di caricamento di un programma.

```
0 REM ****************************
1 REM *AUTORUN CON MODIFICA DEL *
2 REM *VETTORE INTERRUPT PER OT-*
3 REM *TENERE ANIMAZIONI DURANTE*
4 REM *IL CARICAMENTO *MAG-1989 *
5 REM * GIANLUCA VENTURINI SOFT *
6 REM *   SOLO PER COMMODORE 64  *
7 REM *   DOTATO DI  DRIVE 1541  *
8 REM ****************************
10 OPEN15,8,15,"I0":GOSUB1000:REM PRESENTAZIONE
20 GOSUB1100:REM ROUTINE LM *COPIA*
30 GOSUB1200:REM INPUT NOME FILE
40 GOSUB1250:REM INPUT SCELTA
50 GOSUB1300:REM OPEN FILES
60 GOSUB1550:REM ANTIRESET+RUN
70 GOSUB1650:REM COMPLETA
80 PRINT:PRINT"FATTO.":END
90 :
900 INPUT#15,E,E$,TR,SE:IFE<20THENRETURN
910 PRINT"■■ERRORE:■"E;E$;TR;SE:CLOSE15:END
990 :
1000 PRINTCHR$(147)CHR$(5):POKE53280,6:POKE53281,6
1010 PRINT"AUTORUN CON ALTERAZIONE DEL VETTORE "
1020 PRINT"DI INTERRUPT PER OTTENERE ANIMAZIONI"
1030 PRINT"INDIPENDENTI DURANTE IL CARICAMENTO"
1040 PRINT"■-QUESTO ESEMPIO VISUALIZZA QUATTRO     SPRITES IN SEQUENZA";
1050 PRINT" (OMINO CHE CAMMINA)-"
1060 PRINT"■■INSERISCI IL DISCO COL FILE A CUI DARE  QUESTA CARATTERISTICA ";
1070 PRINT"POI PREMI UN TASTO■":POKE198,0:WAIT198,1:POKE198,0
1080 RETURN
1090 :
1100 FORI=0TO31:READA:POKE49152+I,A:NEXT:RETURN
1110 DATA162,5,32,198,255,32,207,255
1120 DATA72,165,144,133,2,162,6,32,201
1130 DATA255,104,32,210,255,165,2,208
1140 DATA3,76,0,192,76,204,255
1190 :
```

```
1200 PRINTCHR$(147):INPUT"NOME PROGRAMMA SORG.";A$:IFA$=""THENGOTO1200
1210 A$=LEFT$(A$,16):B$="AI/"+LEFT$(A$,13)
1220 B$=B$+",P,W":PRINT
1230 RETURN
1240 :
1250 INPUT"ANTI RESET (S/N)";R$:IF(R$<>"S")AND(R$<>"N")THENPRINT"]]":GOTO1250
1260 AR=0:IFR$="S"THENAR=1
1270 RETURN
1290 :
1300 OPEN5,8,5,A$:GET#5,Z$,Z$
1310 OPEN6,8,6,B$
1320 PRINT#6,CHR$(161);:PRINT#6,CHR$(2);:REM PUNTO E VIRGOLA
1330 GOSUB 900:REM RILEVA ERRORI DISCO
1340 I=672
1350 I=I+1:READA:IFA<0THENRETURN
1360 PRINT#6,CHR$(A);:GOTO1350:REM    ';'

1370 REM****ROUTINES LM DI INTERRUPT****
1380 DATA120,32,68,229,162,0,169,6,142,32,208,141,33,208,157,0,216,232,208,250
1390 DATA169,244,141,37,208,169,240,141,38,208,169,241,141,39,208
1400 DATA141,28,208,169,50,141,0,208,169,221,141,1,208,169,16,141,248,7
1410 DATA162,9,189,233,2,157,180,7,202,16,247,169,243,141,20,3,76,49,234
1420 DATA12,15,1,4,9,14,7,46,46,46:REM                                "LOADING..."
1430 DATA120,173,255,4,240,3,76,49,234,76,52,3,0
1439 REM VETTORI
1440 DATA102,3,102,3,124,165,26,167,228,167,134,174,230,40,25,49,76
1450 DATA72,178,0,161,2
1460 DATA102,3
1465 DATA71,254,74,243,145,242,14,242,80,242,51,243,87,241
1470 DATA102,3,234,246
1480 DATA62,241,47,243,102,254,46,223,237,245
1489 REM INTERRUPT/2
1490 DATA169,1,141,21,208,169,70,141,20,3,169,3,141,21,3,76,49,234
1499 REM INTERRUPT/2
1500 DATA120,162,64,160,0,200,208,253,202,16,248
1510 DATA173,248,7,201,19,240,6,238,248,7,76,49,234,169,16,141,248,7,76,49,234
1519 REM RIPRISTINO
1520 DATA120,169,49,141,20,3,169,234,141,21,3,169,0,141,21,208,169,6,141,32,208
1530 DATA169,202,141,38,3,169,241,141,39,3,-1
1540 :
1550 READA:IFA<0THEN1570
1560 IFARTHENPRINT#6,CHR$(A);:I=I+1:GOTO1550: REM
1565 GOTO1550
1570 READA:IFA<0THENRETURN
1580 PRINT#6,CHR$(A);:I=I+1:GOTO1570: REM
1590 :
1599 REM ANTI RESET
1600 DATA169,255,133,55,133,51,169,127,133,56,133,52
1610 DATA169,102,141,0,128,141,2,128,169,3,141,1,128,141,3,128
1620 DATA162,5,189,15,253,157,3,128,202,208,247,-1
1629 REM RUN IN LM
1630 DATA32,89,166,169,0,141,0,8,76,174,167,-1
1640 :
1650 FORJ=ITO1023:PRINT#6,CHR$(0);:NEXT:REM ';'
1660 FORI=0TO255:READA:PRINT#6,CHR$(A);:NEXT:REM ';' **SPRITES**
1670 FORI=1280TO1971:PRINT#6,CHR$(32);:NEXT
1680 FORI=0TO9:READA:PRINT#6,CHR$(A);:NEXT:REM SCRITTA/2
1690 FORI=1982TO2039:PRINT#6,CHR$(32);:NEXT:PRINT#6,CHR$(16);:REM ';'
1700 FORI=2041TO2048:PRINT#6,CHR$(0);:NEXT
1710 SYS49152:REM COPIA IL FILE
```

```
1720 CLOSE5:CLOSE6
1730 GOSUB900:CLOSE15:RETURN
1739 REM DATI SPRITES
1740 DATA 0,0,0,0,242,128,3,250
1750 DATA 160,3,250,160,15,249,128,31
1760 DATA 255,168,61,254,144,61,253,128
1770 DATA 61,252,128,7,160,224,19,160
1780 DATA 0,20,160,0,21,40,0,5
1790 DATA 40,0,5,74,1,1,74,5
1800 DATA 0,82,21,1,82,148,21,66
1810 DATA 176,87,0,160,250,160,128,255
1820 DATA 0,242,128,3,250,160,3,250
1830 DATA 160,15,249,128,31,255,168,63
1840 DATA 126,144,63,125,128,63,124,224
1850 DATA 7,232,0,19,232,0,20,104
1860 DATA 1,21,10,5,5,10,5,5
1870 DATA 90,5;1,82,148,1,82,148
1880 DATA 1,64,160,5,64,168,21,252
1890 DATA 32,60,0,0,42,128,0,255
1900 DATA 0,0,0,0,242,128,3,250
1910 DATA 160,3,250,160,15,249,128,31
1920 DATA 255,168,61,254,144,61,253,128
1930 DATA 61,252,128,7,160,224,19,160
1940 DATA 0,20,160,0,21,32,0,5
1950 DATA 40,0,5,74,1,1,74,5
1960 DATA 5,66,21,21,18,148,20,82
1970 DATA 176,80,64,160,170,252,128,255
1980 DATA 0,242,128,3,250,160,3,250
1990 DATA 160,15,249,128,31,255,168,63
2000 DATA 126,144,63,125,128,63,124,224
2010 DATA 7,232,0,19,232,0,20,104
2020 DATA 1,21,10,5,5,74,5,1
2030 DATA 74,5,0,82,148,0,82,148
2040 DATA 1,64,160,5,0,168,21,0
2050 DATA 32,42,128,0,15,240,0,0
2059 REM SCRITTA/2
2060 DATA 12,5,7,7,9,39,3,3,3,39
2070 END
```

```
1 ******************************
2 * Disassemblato -1- routine *
3 *         copia file         *
4 ******************************
5 INIZIO    LDX   #$05    ; dichiara il file
6           JSR   $FFC6   ; n. 5 di input
7           JSR   $FFCF   ; preleva un byte
8           PHA           ; lo salva nello stack
9           LDA   #$90    ; legge lo status
10          STA   $02     ; lo salva
11          LDX   #$06    ; dichiara il file
12          JSR   $FFC9   ; n. 6 di output
13          PLA           ; recupera il byte
14          JSR   $FFD2   ; lo invia in output
15          LDA   $02     ; recupera lo status
16          BNE   FINE    ; se il file
16          JMP   INIZIO  ; non e' terminato ripeti
17 FINE     JMP   $FFCC   ; altrimenti azzera i canali di I/O.
```

```
1 ******************************
2 * Disassemblato -2- routine *
3 *       interrupt / 1        *
4 ******************************
5 INIZIO    SEI             disabilita gli interrupts
```

```
 6              JSR    $E544    ; pulisce lo schermo
 7              LDX    #$00     ; setta
 8              LDA    #$06     ; i colori
 9              STX    $D020    ; del bordo
10              STA    $D021    ; e di sfondo
11 LOOP1        STA    $D800,X  ; riempie la prima pagina
12              INX             ; della memoria colore
13              BNE    LOOP1    ; del colore di sfondo
14              LDA    #$F4     ; setta
15              STA    $D025    ; i valori
16              LDA    #$F0     ; per il
17              STA    $D026    ; multicolore
18              LDA    #$F1     ; e per il colore
19              STA    $D027    ; dello sprite
20              STA    $D01C    ; attiva flag multicolor
21              LDA    #$32     ; setta
22              STA    $D000    ; posizione X
23              LDA    #$DD     ; e posizione
24              STA    $D001    ; Y dello sprite
25              LDA    #$10     ; dati dello sprites
26              STA    $07F8    ; allocati da 1024 in poi
27              LDX    #$09     ; scrive
28 LOOP2        LDA    MSG1,X   ; il messaggio
29              STA    $07B4,X  ; in basso
30              DEX             ; sullo schermo
31              BPL    LOOP2    ;
32              LDA    #<ROUT2  ; modifica vettore interrupt
33              STA    $0314    ; verso la seconda routine
34              JMP    $EA31    ; esegue normale ciclo interrupt
35
36 MSG1         .TXT   'LOADING...'
37 ******************************
38 * Disassemblato -3- routine *
39 *        interrupt / 2       *
40 ******************************
41 ROUT2        SEI             ; disabilita gli interrupts
42              LDA    $04FF    ; attende che siano caricati tutti i
43              BEQ    JUMP1    ; dati dello sprite nella memoria video
44              JMP    $EA31    ; se no: esegue ciclo interrupt normale
45 JUMP1        JMP    ROUT3    ; altrimenti salta alla terza routine
46

47 ******************************
48 * Disassemblato -4- routine *
49 *        interrupt / 3       *
50 * (non contigua alla ROUT2) *
51 ******************************
52 ROUT3        LDA    #$01     ; abilita
53              STA    $D015    ; lo sprite
54              LDA    #<ROUT4  ; modifica il vettore
55              STA    $0314    ; di interrupt
56              LDA    #>ROUT4  ; verso la
57              STA    $0315    ; quarta routine
58              JMP    $EA31    ; esegue interrupt solito
59
60 ******************************
61 * Disassemblato -5- routine *
62 *        interrupt / 4       *
63 ******************************
64 ROUT4        SEI             ; disabilita gli interrupts
65              LDX    #$40     ; crea un
66 LOOP3        LDY    #$00     ; certo ritardo
67 LOOP4        INY             ; tra un
68              BNE    LOOP4    ; movimento e
69              DEX             ; l ' altro
70              BPL    LOOP3    ; dello sprite
71              LDA    $07F8    ; se lo sprite ha gia'
72              CMP    #$13     ; eseguito il quarto 'movimento'
73              BEQ    JUMP2    ; vai a JUMP2
74              INC    $07F8    ; altrimenti prossimo movinemto
```

```
75            JMP   $EA31    ; esegue ciclo interrupt
76 JUMP2      LDA   #$10     ; sprite nella
77            STA   $07F8    ; posizione iniziale
78            JMP   $EA31    ; esegue ciclo interrupt
79
80 ******************************
81 * Disassemblato -6- routine *
82 *     ripristino interrupt    *
83 ******************************
84 RIPRIS     SEI            ; disabilita gli interrupts
85            LDA   #$31     ; ripristina
86            STA   $0314    ; i valori
87            LDA   #$EA     ; normali
88            STA   $0315    ; del vettore interrupt
89            LDA   #$00     ; disabilita
90            STA   $D015    ; lo sprite
91            LDA   #$06     ; e setta il colore
92            STA   $D020    ; di bordo
93            LDA   #$CA     ; ripristina
94            STA   $0326    ; i valori
95            LDA   #$F1     ; normali
96            STA   $0327    ; del vettore output
97
```

```
 98 ******************************
 99 * Disassemblato -7- routine *
100 *    antireset (opzionale)   *
101 ******************************
102 ARESET     LDA   #$FF      ; setta
103            STA   $37       ;  la
104            STA   $33       ; fine
105            LDA   #$7F      ; della
106            STA   $38       ; memoria
107            STA   $34       ; a 32767
108            LDA   #$66      ; in caso
109            STA   $8000     ; di reset
110            STA   $8002     ; o runstop-restore
111            LDA   #$03      ; viene
112            STA   $8001     ; eseguita
113            STA   $8003     ; la routine di run
114            LDX   #$05      ; vengono
115 LOOP5      LDA   $FD0F,X   ; ricopiati
116            STA   $8003,X   ; i caratteri
117            DEX             ; 'CBM80'
118            BNE   LOOP5     ; dalla ROM a 32772 e segg.
119
120 ******************************
121 * Disassemblato -8- routine *
122 *         di RUN in L.M.      *
123 ******************************
124 RUN        JSR   $A659     ; istruzioni
125            LDA   #$00      ; per lanciare
126            STA   $0800     ; il
127            JMP   $A7AE     ; programma
128
129 ******************************
```

a cura di **Andrea Ciaramella**

PROJECT FIRESTART (Commodore 64)

Siete fans dei film di Dario Argento? Se la vostra risposta è sì, *Project Firestart* è il gioco che fa per voi. Siete non-violenti convinti?

Allora girate pagina. Infatti giocando *Project Firestart* avrete a che fare con momenti di intensa violenza.

Ma andiamo per gradi, cominciando col raccontare la storia: siamo nell'anno 2033, da qualche parte dello spazio. Da alcuni giorni Prometeus (una base spaziale incaricata di ricerche di genetica per creare nuovi esseri capaci di lavorare in condizioni critiche) non risponde più alle chiamate radio ed è quindi richiesto il vostro intervento; siete inviati all'interno del Prometeus per accertamenti. Partendo da questa trama i programmatori della *Electronic Arts* hanno realizzato un gioco che passerà alla storia. Sulla base spaziale farete incontri terrificanti; l'intero gioco è pieno di schermate sanguinolente (del tipo di quelle incontrate nelle avventure di Rod Pike) che lo rendono, per alcuni, appassionante.

Lo scopo del gioco è quello di innescare l'autodistruzione di Prometeus e fuggire con una navicella spaziale. L'impresa deve essere portata a termine nel giro di due ore.

Parlando dell'aspetto tecnico, il gioco è coinvolgente e gli intermezzi che appaiono di tanto in tanto contribuiscono a renderlo ancora più reale.

La quantità dei luoghi visitati, e la loro implementazione grafica, sono veramente impressionanti. Il sonoro è quasi del tutto assente, anche se questo particolare non modifica la suspance che permea l'intero gioco.

L'inaudita, pura violenza su cui si basa il videogame non sarà certamente gradita a tutti, ma forse può contribuire a sfogare i propri istinti su un joystick piuttosto che nella realtà quotidiana. Etica a parte, il gioco merita almeno di essere esaminato, ma attenti: quando lo caricate, tenete le luci accese e, se possibile, invitate molti amici; la paura si sopporta meglio quando si è in tanti.

THE RUNNING MAN (Amiga)

Il film *The running man*, protagonista Arnold Schwarzenegger, parla di un uomo accusato ingiustamente di omicidio; per riscattare la sua pena viene costretto a compiere un corsa contro tutti per raggiungere la libertà.

Perchè vi raccontiamo la trama di un film? Non certo perchè vogliamo rendervi partecipi delle novità dello schermo, ma solo perchè *The running man* è diventato un gioco per Amiga. La trama è identica a quella del film e lo scopo è quello di portare a termine la vostra corsa.

Appena terminato il caricamento iniziale apparirà una stupefacente presentazione con immagini tratte direttamente dal film, accompagnata da ottimi effetti di sintesi vocale.

Il programma mantiene un'alta qualità grafica anche durante il gioco vero e proprio (lo scrolling, tuttavia, potevano renderlo migliore) e gli effetti sonori sono piuttosto curati.

La facilità con cui operare le varie mosse, però, diminuisce il livello qualitativo del prodotto (anche se non di molto) e la sua longevità, di conseguenza, è... breve. Il trucco si riduce a memorizzare le varie mosse; in seguito il più è fatto. *The running man* potrebbe piacervi ma, per evitare delusioni, è bene comprarlo solo dopo averlo visto in funzione.

In conclusione, si tratta del classico gioco che serve agli Amighisti per sbalordire gli amici (che magari hanno ancora un C/64) grazie agli effetti grafici e sonori realmente spettacolari; l'eccessiva facilità con cui la missione viene conclusa, invece, deve essere considerata un difetto.

POPULOUS (Amiga)

Avete mai pensato a come potessero sentirsi gli dei della Grecia, Zeus in testa, quando, per accaparrarsi fedeli e devoti, dovevano sottoporsi a lotte intestine?

In *Populous* diventate un dio e, come tutti gli dei che si rispettino, dovete cercare di favorire l'espansione territoriale dei vostri fedeli a danno del dio avversario. Per realizzare il vostro scopo userete tutti i poteri a disposizione, che comprendono la capacità di provocare eruzioni, terremoti, di creare terra, di farla scomparire e altre diavolerie (o, se preferite, "deate") del genere.

E' possibile giocare da soli contro il computer o contro un amico (o forse dovremmo chiamarlo nemico?); questi può essere presente fisicamente al vostro fianco, oppure in contatto telefonico via modem.

In *Populous* sono presenti diversi mondi, scenari dei vari combattimenti: mondi ghiacciati, infuocati o mondi dall'aspetto decisamente terrestre.

Il gioco si basa su di un semplice concetto: più il terreno è pianeggiante, maggiore è la superficie vivibile e maggiore è lo stadio evolutivo del vostro popolo. In una zona montagnosa, quindi, vi saranno pochi abitanti e questi vivranno in capanne; mentre in una zona di pianura gli abitanti saranno più numerosi e vivranno in grandi castelli.

Riteniamo che questo sia uno dei giochi più intelligenti realizzati dai tempi del mitico *Elite* e dicendo questo intendiamo fargli un complimento.

Tecnicamente è realizzato in maniera impeccabile ed è difficile staccarsene se non dopo che aver mandato fuori uso il joystick. Volendo esprimere un giudizio globale in una sola parola, rimanendo in argomento, possiamo definirlo divino.

THE REAL GHOSTBUSTERS (Commodore 64)

Sono passati alcuni anni dal primo, e fino a poco tempo fa, unico, titolo della serie *Ghostbusters*. A quei tempi Ghostbusters fece scalpore sia per la sua realizzazione grafica (e sonora) sia per la concezione stessa del gioco. Possiamo dire lo stesso di *The Real Ghostbusters*? Cerchiamo di vederlo insieme.

Nel gioco si interpreta la parte di un membro della squadra acchiappafantasmi che si aggira per la città (composta di dieci livelli a scorrimento multidirezionale) risucchiando tutte le entità ectoplasmatiche ivi presenti.

The Real Ghostbusters puo essere giocato sia da soli che insieme ad un amico.

Lo schema di gioco è molto simile a quello di *Alien Syndrome* anche se, purtroppo, la grafica non raggiunge il suo stesso livello; sembra piuttosto realizzata non per un C/64, ma per un computer ibrido, a metà strada tra C/64 e Vic 20.

Una implementazione grafica migliore, insomma, sarebbbe riuscita a rendere più accettabile il gioco, ma quelli della *Data East* hanno avuto idee differenti.

Il sonoro contribuisce a rialzare il livello globale e, per quanto riguarda le altre caratteristiche, la difficoltà è abbastanza ben calibrata come anche la vasta gamma di prede da catturare.

Eccettuata la grafica, *The Real Ghostbusters* è in definitiva un buon gioco e sarà un acquisto gradito per gli sfegatati del genere. Se però puntate all'aspetto spettacolare dei giochi, lasciatelo perdere.

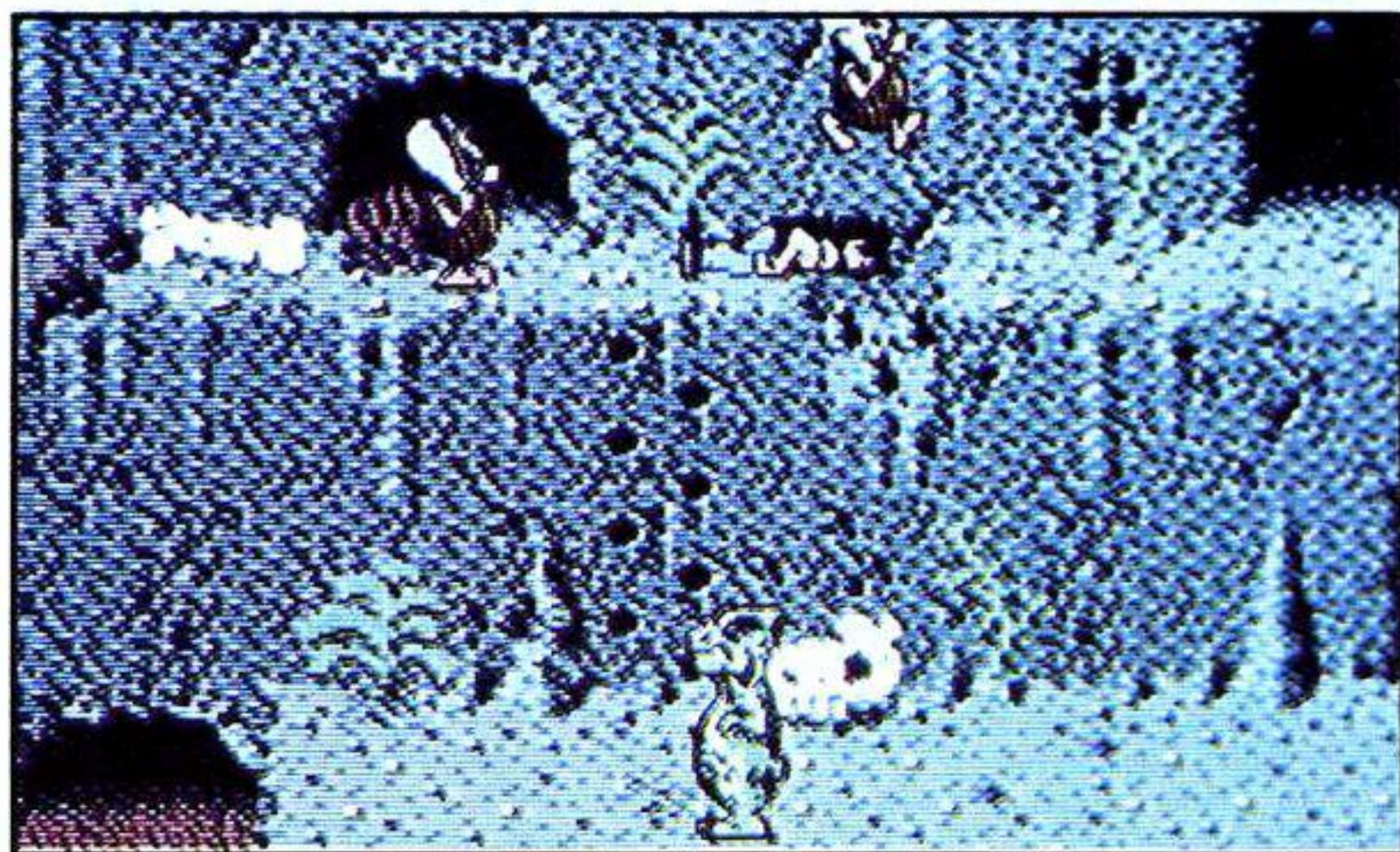
RENEGADE III (Commodore 64)

Il nostro amato *Renegade* torna per la terza volta sui nostri monitors e, dobbiamo dirlo, lo fa alla grande. La struttura del gioco non risulta molto diversa dagli altri (anzi, è identica) ma possiamo assicurarvi che *Renegade III* ha tutto quello che mancava agli altri due titoli.

Ed ecco la trama (provate un po' a indovinare?). Hanno rapito la ragazza di Renegade (originale, vero?) ma questa volta, grazie ad uno scherzo giocatogli dai suoi nemici, il nostro eroe verrà teletrasportato in diverse epoche: la preistoria, l'Egitto dei tempi dei faraoni, il medioevo e... sorpresa!

Per giocare bisogna scontrarsi in risse colossali con cavernicoli, dinosauri, draghi, mummie e chi più ne ha più ne metta. Parlando della realizzazione tecnica, possiamo dire che *Renegade III* ha una grafica alquanto spettacolare, un sonoro con musiche molto valide e i movimenti vengono animati in maniera decisamente veloce.

Le mosse a disposizione di Renegade sono piuttosto varie, ma la difficoltà è calibrata in maniera non sempre consona a quelle che sono le capacità di un giocatore medio. Comunque, se la struttura ripetitiva del gioco (calci, sberle e pugni) non vi infastidisce, *Renegade III* si rivelerà sicuramente un acquisto azzeccato e, con i suoi quattro livelli, riuscirà a trattenervi per molto tempo.

Se, invece, siete abituati a giocare con giochi della Infocom e i picchiaduro non vi attirano, questo è il momento di cambiare opinione.

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 28.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 615.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 8.930.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3.2.1 - Interprete GW-Basic

PC60/40C - L. 9.180.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.965.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 2.300.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.715.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 3.050.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000

Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000

Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000

Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000

Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000

Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D

1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64
Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64
Permette il collegamento a Videotel, P.G.E. e banche dati - **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000

Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000

Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000

Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000

Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000

Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000

Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000

Nastro per stampante

Stampante MPS 1500C - L. 550.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000

Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

LOMBARDIA

Milano
• AL RISPARMIO - V.LE MONZA 204
• BCS - VIA MONTEGANI 11
• BRAHA A. - VIA PIER CAPPONI 5
• E.D.S. - C.SO PORTA TICINESE 4
• FAREF - VIA A. VOLTA 21
• FLOPPERIA - V.LE MONTENERO 31
• GBC - VIA CANTONI 7 - VIA PETRELLA 6
• GIGLIONI - V.LE LUIGI STURZO 45
• L'UFFICIO 2000 - VIA RIPAMONTI 213
• LOGITEK - VIA GOLGI 60
• LU - MEN - VIA SANTA MONICA 3
• MARCUCCI - VIA F.LLI BRONZETTI 37
• MELCHIONI - VIA P. COLLETTA 37
• MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
• NEWEL - VIA MAC MAHON 75
• PANCOMMERZ ITALIA - VIA PADOVA 1
• SUPERGAMES - VIA VITRUVIO 38
• 68000 E DINTORNI - VIA WASHINGTON 91

Provincia di Milano
• GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
• F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
• TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
• OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
• AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
• GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
• CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
• PENATI - VIA VERDI 28/30 - CORBETTA
• EPM SYSTEM - V.LE ITALIA 12 - CORSICO
• P.G. OSTELLARI - VIA MILANO 300 - DESIO
• CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
• COMPUTEAM - VIA VECELLIO 41 - LISSONE
• M.B.M. - C.SO ROMA 112 - LODI
• L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
• BIT 84 - VIA ITALIA 4 - MONZA
• IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
• I.C.O. - VIA DEI TIGLI 14 - OPERA
• R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
• ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
• TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
• NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
• COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
• ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
• IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE

Bergamo
• D.R.B. - VIA BORGO PALAZZO 65
• TINTORI ENRICO &C. - VIA BROSETA 1
• VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO

Provincia di Bergamo
BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
• COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
• B M R - VIA BUTTARO 4/T - DALMINE
• MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
• OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
• COMPUTER POINT - VIA LANTIERI 52 - SARNICO
• A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO

Brescia
• MASTER INFORMATICA - VIA F.LLI UGONI 10/B

PROVINCIA DI BRESCIA
• MISTER BIT - VIA MAZZINI 70 - BRENO
• CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
• VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
• MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
• BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
• INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
• "PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21

Como
• IL COMPUTER - VIA INDIPENDENZA 90
• 2M ELETTRONICA - VIA SACCO 3

Provincia di Como
• ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
• DATA FOUND - VIA A. VOLTA 4 -ERBA
• CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
• FUMAGALLI - VIA CAIROLI 48 -LECCO
• RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO

Cremona
• MONDO COMPUTER - VIA GIUSEPPINA 11/B
• PRISMA - VIA BUOSO DA DOVARA 8
• TELCO - P.ZZA MARCONI 2/A

Provincia di Cremona
• ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
• EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA

Mantova
• COMPUTER CANOSSA - GAL. FERRI 7
• 32 BIT - VIA C. BATTISTI 14
• ELET. di BASSO - V.LE RISORGIMENTO 69

Provincia di Mantova
• CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO

Pavia
• POLIWARE - C.SO C. ALBERTO 76
• SENNA GIANFRANCO - VIA CALCHI 5

Provincia di Pavia
• A. FERRARI - C.SO CAVOUR 57 - MORTARA
• LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
• M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO

Sondrio
• CIPOLLA MAURO - VIA TREMOGGE 25

Provincia di Sondrio
• FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO

Varese
• ELLE - EFFE - VIA GOLDONI 35
• IL C.TRO ELET. - VIA MORAZZONE 2
• SUPERGAMES - VIA CARROBBIO 13

Provincia di Varese
• BUSTO BIT - VIA GAVINANA 17-BUSTO A.
• MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
• PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
• GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
• J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE

Alessandria
• BIT MICRO - VIA MAZZINI 102
• SERV. INFOR. - VIA ALESSANDRO III 47

Provincia di Alessandria
• SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
• SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
• COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA

Asti
• ASTI GAMES - C.SO ALFIERI 26
• RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)

Cuneo
• ROSSI COMPUTERS - C.SO NIZZA 42

Provincia di Cuneo
• PUNTO BIT - C.SO LANGHE 26/C - ALBA
• BOSETTI - VIA ROMA 149 - FOSSANO'
• COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO

Novara
• PROGRAMMA 3 - V.LE BUONARROTI 8
• PUNTO VIDEO - C.so RISORGIMENTO 39/B

Provincia di Novara
• COMPUTER - VIA MONTE ZEDA 4 -ARONA
• ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
• S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
• ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
• TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA

Torino
• ABA ELETTRONICA - VIA C. FOSSATI 5/P
• ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
• COMPUTER HOME - VIA SAN DONATO 46/D
• COMPUTING NEW - VIA M. POLO 40/E
• C.D.M. ELETTR. - VIA MAROCHETTI 17
• DE BUG - C.SO V. EMANUELE II 22
• DESME UNIVERSAL - VIA S.SECONDO 95
• FDS ALTERIO - VIA BORGARO 86/D
• IL COMPUTER - VIA N. FABRIZI 126
• MICRONTEL - C.SO D. degli ABRUZZI 28
• PLAY GAMES SHOP - VIA C. ALBERTO 39/E
• RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
• SMT ELETTRONICA - VIA BIBIANA 83/bis

Provincia di Torino
• PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
• BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
• HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
• MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
• I.C.S. - VIA TORINO 73 - IVREA
• DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
• EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
• DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
• FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
• GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE

Vercelli
• ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
• ELETTRONICA - STRADA TORINO 15

Provincia di Vercelli
• C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
• SIGEST - VIA BERTODANO 8 - BIELLA
• REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
• FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
• STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO

Belluno
• UP TO DATE - VIA V. VENETO 43

Provincia di Belluno
• GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE

Padova
• BIT SHOP - VIA CAIROLI 11
• COMPUMANIA - VIA T. CAMPOSANPIERO 37
• D.P.R. DE PRATO R. - V.LO LOMBARDO 4
• G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
• SARTO COMPUTER - VIA ARMISTIZIO 79

Provincia di Padova
• COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA

Treviso
• BIT 2000 - VIA BRANDOLINI D'ADDA 14
• GUERRA EGIDIO & C. - V.LE CAIROLI 95

Provincia di Treviso
• DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
• SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
• FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL

Venezia
• GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
• TELERADIO FUGA - SAN MARCO 3457

Provincia di Venezia
• GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
• REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE

Verona
• CASA DELLA RADIO - VIA CAIROLI 10
• TELESAT - VIA VASCO DE GAMA 8

Provincia di Verona
• UBER - CP 0363(RAG.SOC. DERTA) - VIA MASCAGNI 31 -CASTEL D'AZZANO
• FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
• COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA

Vicenza
• ELET. BISELLO - V.LE TRIESTE 427/429
• SCALCHI MARKET - VIA CA' BALBI 139

Provincia di Vicenza
• SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
• GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA

Gorizia
• E.CO. ELETTRONICA - VIA F.LLI COSSAR 23

Trieste
• AVANZO GIACOMO - P.ZZA CAVANA 7
• COMPUTER SHOP - VIA P. RETI 6
• COMPUTIGI - VIA XX SETTEMBRE 51
• CTI - VIA PASCOLI 4

Udine
• MOFERT 2 - VIA LEOPARDI 21
• R.T. SISTEM UDINE - VIA L. DA VINCI 99

Provincia di Udine
• IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
• IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE

Bolzano
• COMPUTER POINT - VIA ROMA 82/A
• MATTEUCCI PRESTIGE - VIA MUSEO 54

Provincia di Bolzano
• RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
• ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
• ERICH KONTSCHIEDER - PORTICI 313 - MERANO
• ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO

Trento
• CRONST - VIA G. GALILEI 25

Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA' 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL'UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA'DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G.FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF. - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R 2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI -
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 - PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090

SYSTEMS EDITORIALE PER TE

La voce III
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far *parlare* il computer, sia di farlo *Cantare!* Diversi esempi allegati.
Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.
Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.
Cassetta: L. 12000
Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).
Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese ed i guadagni di una famiglia.
Cassetta: L. 12000
Disco: L. 12000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.
Cassetta: L. 12000
Disco: L. 12000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.
Cassetta: L. 20000
Disco: L. 20000

Analisi di Bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.
Cassetta: L. 20000
Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 ed i rudimenti di programmazione. Interattivo.
Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.
Cassetta: L. 12000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore. Diversi esempi allegati.
Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.
Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.
Solo su disco: L. 25000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive. Ideale per studenti.
Disco: L. 25000

L.M. + Routine grafiche
Un fascicolo speciale (corredato di dischetto) suddiviso in due parti: corso completo di linguaggio macchina 6502 e implementazione di numerose routine che aggiungono al C/64 istruzioni Basic specifiche per la grafica, comprese quelle per disegnare in prospettiva!
Fascicolo + disco: L. 16000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.
Disco: L. 15000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.
Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.
Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su *"Commodore Computer Club"*. In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro. Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.
Ogni dischetto: L. 12.000

LIBRI TASCABILI

64 programmi per il C/64
Raccolta di programmi (giochi e utilità) semplici da digitare e da usare. Ideale per i principianti. ***(126 pag.)***
L. 4800

I miei amici C/16 e Plus/4
Il volumetto, di facile apprendimento, rappresenta un vero e proprio mini-corso di Basic per i due computer Commodore. Numerosi programmi, di immediata digitazione, completano la parte teorica. ***(127 pag.)***
L. 7000

62 programmi per C/16, Plus/4
Raccolta di numerosissimi programmi, molto brevi e semplici da digitare, per conoscere più a fondo il proprio elaboratore. Ideale per i principianti. ***(127 pag.)***
L. 6500

Micro Pascal 64
Descrizione accurata della sintassi usata dal linguaggio Pascal "classico". Completa il volume un programma di emulazione del PL/0 sia in formato Microsoft sia in vesione C/64 (da chiedere, a parte, su disco). ***(125 pag.)***

L. 7000

Dal Registratore al Drive
Esame accurato delle istruzioni relative alle due più popolari periferiche del C/64. Diversi programmi applicativi ed esempi d'uso. ***(94 Pag.)***

L. 7000

Il linguaggio Pascal
Esame approfondito della sintassi usata nel famoso compilatore. ***(112 pag.)***

L. 5000

Utilities e giochi didattici
Raccolta di numerosi programmi, in versione C/64 e Spectrum, di particolare interesse per chi intenda svliluppare software didattico. ***(127 pag.)***

L. 6500

Simulazioni e test per la didattica
Raccolta di numerosi programmi che approfondiscono e tendono a completare la trattazione già affrontata sul precedente volume. *(127 pag.)*

L. 7000

Dizionario del Personal Computer
Raccolta dei termini più diffusi nel campo professionale: dizionario inglese - italiano. (Edizione ridotta). ***(96 pag.)***

L. 8000

Dizionario dell'Informatica
Dizionario inglese italiano di tutti i termini usati nell'informatica. (Edizione completa). ***(385 pag.)***

L. 20000

Word processing: istruzioni per l'uso
Raccolta delle principali istruzioni dei più diffusi programmi di w/p per i sistemi Ms-Dos: Word Star, Samna, Multimate Advantage, Word 3. ***(79 pag.)***

L. 5000

Telefax
Volumetto divulgativo sull'importanza del Telefax e sulle sue modalità operative caratteristiche. ***(66 pag.)***

L. 5000

Come compilare un giornale aziendale in Azienda
I principali problemi per chi opera in ambiente DPT, affrontati e risolti con la massima chiarezza e semplicità. ***(80 pag.)***

L. 5000

Unix
Un volumetto per saperne di più sul sistema operativo professionale per eccellenza. ***(91 pag.)***

L. 5000

ABBONAMENTO

Commodore Computer Club

11 fascicoli: L. 50.000

ARRETRATI

Ciascun numero arretrato

di C.C.C. L. 5000

COME RICHIEDERE I PRODOTTI SYSTEMS

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, L. 3500 per spese di imballo e spedizione, oppure L. 6000 se si desidera la spedizione per mezzo raccomandata.
Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L. 50000.

Per gli ordini, compilare un normale modulo di C/C postale indirizzato a:

C/C Postale N. 37 95 22 07
Systems Editoriale
Viale Famagosta, 75
20142 MILANO

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche i prodotti desiderati ed il tipo di spedizione da effettuare.

Per sveltire la procedura di spedizione sarebbe opportuno inviare, a parte, una lettera riassuntiva dell'ordine effettuato, allegando una fotocopia della ricevuta del versamento.

Volendo una spedizione in contrassegno è necessario anticipare la cifra di L. 10000 (diecimila), da inviare secondo le modalità prima indicate, indipendentemente dalla quantità di materiale richiesto, e da conteggiare, comunque, IN AGGIUNTA alla cifra risultante dall'ordine. (Si sconsiglia, pertanto, la richiesta di di prodotti in contrassegno)

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare, oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a:

Systems Editoriale
Milano